

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Mise en oeuvre d'une relation EDI: aspects méthodologiques et techniques

Charlier, Laurent; Hubermont, Jean-François

Award date:
1995

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année académique 1994-1995

Mise en oeuvre d'une relation EDI:
aspects méthodologiques et techniques.

*Laurent Charlier,
Jean-François Hubermont*

Promoteur: Ph. van Bastelaer

Mémoire présenté en vue de l'obtention du grade de Licencié et Maître en
Informatique.

*Nous remercions vivement toutes les personnes qui ont de près
ou de loin contribué à la réalisation de ce travail.*

En particulier:

Monsieur Ph. van Bastelaer qui en a assuré la direction;

*Madame V. Nachtergaele-Hogne, Monsieur P. Waterkeyn
et les membres de la CITA qui nous ont témoigné de leur disponibilité;*

*Craig Parker et Darjan Petric qui furent des partenaires particulièrement amicaux
et sans lesquels ce travail n'aurait pu être réalisé.*

*Derniers remerciements, et non des moindres, à nos parents
sans lesquels ces études n'auraient pas été possibles.*

TABLE DES MATIERES

ABSTRACT:	1
INTRODUCTION	2
CHAPITRE 1. LE PHENOMENE EDI	3
1.1. Naissance du concept d'EDI	3
1.2. Définition de l'EDI	3
1.3. Nécessité de standards pour la mise en oeuvre de l'EDI	5
1.3.1. Niveaux de standardisation	5
1.3.2. Avantage des standards de représentation des données	7
1.3.3. Standards de communication	8
1.4. Présentation générale du standard UN/EDIFACT	9
1.5. Autres types de standards de représentation	10
1.6. Les contextes d'intégration de l'EDI	11
1.6.1. Le système papier	12
1.6.2. Intégration avec la base de données existante de l'organisation	14
1.6.3. Intégration à des applications et à des processus existants	15
1.6.4. Développement de nouvelles applications et de nouveaux processus basés sur l'EDI	16
1.6.5. Développement d'un système qui intègre complètement les applications à l'EDI	17
1.6.6. Conclusion et remarques	18
CHAPITRE 2. PORTEE DU TRAVAIL	21
2.1. Contexte et objectifs généraux	21
2.1.1. Simulation d'échanges EDI Curtin/Maribor	21
2.1.1.1. Système EDI élaboré à Curtin	22
2.1.1.2. Mise en oeuvre de l'EDI à Maribor	23
2.1.2. Objectifs par rapports à la simulation	24
2.2. Structure de l'exposé	26
CHAPITRE 3: LA NORME EDIFACT ET SES COMPOSANTES	27
3.1. Les jeux de caractères et les deux niveaux de syntaxe	28
3.2. Les éléments de données	29
3.2.1. Les éléments de données (simples)	29
3.2.2. Les éléments de données composites	29
3.2.3. Données codées	30
3.2.4. Qualifiants et données qualifiées	30
3.3. La notion de segments de données EDIFACT	30
3.3.1 Structure d'un segment	31
3.3.2. Les segments de contrôle et la structure de l'interchange	32
3.3.2.1. Le segment UNA	33

3.2.2.2. Les segments UNB et UNZ	33
3.3.2.3. Les segments UNG et UNE:	34
3.3.2.4. Les segments UNH, UNT et UNS	34
3.4. La structure des messages EDIFACT	35
3.4.1. Description du diagramme d'un message	36
3.4.1.1. Emboîtement vertical des segments	36
3.4.1.2. Groupage des segments	36
3.4.2. Les segments de contrôle d'un message	38
3.4.2.1. Le segment UNH	38
3.4.2.2. Le segment UNT	38
3.4.2.3. Les segments UNS	38
3.5. La syntaxe de transfert des fichiers d'interchange EDIFACT	39
3.5.1. Codage d'un message	39
3.5.2. Codage d'un segment et règles de compression dans le codage et la transmission d'un segment	39
CHAPITRE 4. ANALYSE DES CHOIX A EFFECTUER POUR METTRE EN OEUVRE DES ECHANGES EDI	43
4.1. Choix des partenaires	43
4.2. Evaluation des bénéfices apportés par l'EDI	44
4.3. Choix d'un niveau d'intégration de l'EDI	44
4.4. Estimation des investissements nécessaires à la mise en oeuvre de l'EDI	44
4.5. Choix des standards EDI	45
4.6. Choix des documents à échanger par l'EDI	45
4.7. Choix d'un logiciel de traduction	45
4.8. Choix du matériel	46
4.9. Choix d'un moyen de télécommunication	46
CHAPITRE 5. FONCTIONNALITES ET CLASSIFICATION DES LOGICIELS EDI	48
5.1. Fonctionnalités des logiciels EDI	48
5.1.1. Les fonctionnalités d'entrée et sortie	49
5.1.2. La conversion	50
5.1.3. La communication interne	51
5.1.4. La communication externe	51
5.1.5. La gestion des événements	51
5.2. Catégories de logiciels EDI	51
5.2.1. Les convertisseurs EDI	52
5.2.2. Les passerelles EDI	52
5.2.3. Les stations de travail EDI	52
CHAPITRE 6. CHOIX D'UN LOGICIEL EDI	54
6.1. Groupe 1: Informations concernant la société qui fournit le logiciel	55

6.2. Groupe 2: L'interaction entre l'utilisateur et l'interface (Screen functionality)	56
6.3. Groupe 3: Les fonctionnalités offertes par le système	56
6.4. Groupe 4: La flexibilité	57
6.5. Groupe 5: Les standards EDI	57
6.6. Groupe 6: Les comptes rendus des opérations effectuées (reporting)	57
6.7. Groupe 7: La communication	58
6.8. Groupe 8: Installation et maintenance	58
6.9. Groupe 9: Le support	58
6.10. Groupe 10: Les coûts	59
CHAPITRE 7. PRESENTATION D'ATLAS EDI	60
7.1. Informations concernant la société qui fournit le logiciel	60
7.2. Interactions entre l'utilisateur et l'interface (Screen functionality)	60
7.3. Fonctionnalités offertes par le système	61
7.4. La flexibilité du logiciel	66
7.4.1. Spécification des paramètres de traduction et association de certains types de messages à chaque partenaire	66
7.4.2. Construction des tables de mapping	67
7.4.3. Configuration du système	68
7.5. Les standards EDI	68
7.6. Comptes rendus des opérations effectuées (reporting)	69
7.7. La communication	69
7.8. Installation et maintenance	69
7.9. Le support	70
CHAPITRE 8. CONFIGURATION DU LOGICIEL	71
8.1. Introduction	71
8.2. Configuration du logiciel ATLAS-EDI	72
8.2.1. Sélection des paramètres à valider pendant la traduction et spécification de la syntaxe et du format des fichiers plats	73
8.2.1.1. Sélection des paramètres de la traduction IMPORT et définition de la structure des fichiers plats importés	74
8.2.1.2. Sélection des paramètres de la traduction EXPORT et spécification de la structure des fichiers plats à exporter	80
8.2.2. Construction d'une table de mapping associée au type de message étudié	83
8.2.3. Edition de la table de mapping associée au type de message	87
8.2.4. Enregistrement du profil des partenaires	92
8.2.4.1. Enregistrement du profil de l'utilisateur	92
8.2.4.2. Enregistrement du profil des partenaires associés à l'utilisateur	94

CHAPITRE 9. ANALYSE DE LA TRADUCTION DE QUELQUES TYPES DE MESSAGES	99
9.1. Traduction d'un interchange EDIFACT en un fichier plat	99
9.2. Traduction d'un fichier plat en un interchange EDIFACT	105
9.3. Traduction d'un interchange comportant plusieurs messages	110
CHAPITRE 10. PRESENTATION DU MAPPER ET COMPARAISON DE SON PROCESSUS DE TRADUCTION AVEC CELUI D'ATLAS	114
10.1. Introduction	114
10.2. Interfaces du Mapper	114
10.3. Les tables de mapping du Mapper	115
10.4. Les fonctionnalités du Mapper	119
10.5. Comparaison Mapper - ATLAS-EDI	120
10.5.1. Les faiblesses d'ATLAS-EDI vis-à-vis du Mapper	120
10.5.2. Les avantages d'ATLAS-EDI vis-à-vis du Mapper	120
10.6. Conclusion	121
CHAPITRE 11. INTERFACES FENETREES ET APPLICATIONS	122
CHAPITRE 12. METHODE DE CONCEPTION DES MESSAGES ECHANGES	126
12.1. Choix des messages échangés avec Craig Parker	126
12.2. La conception des messages EDIFACT définie selon les Nations unies	127
12.2.1. Introduction	127
12.2.2. Principes généraux	127
12.2.3. Analyse des éléments de données	128
12.2.4. Analyse des segments de données	128
12.2.5. Procédure générale d'adoption d'un message	129
CHAPITRE 13. ANALYSE DU CHEMIN SUIVI PAR LES DONNEES	130
13.1. Introduction	130
13.2. Le réseau Internet	130
13.2.1. Introduction	130
13.2.2. La structure du réseau Internet	130
13.2.3. Les protocoles TCP/IP	130
13.2.4. Le protocole SMTP	132
13.2.5. L'encapsulation MIME d'objets EDI	134
13.2.5.1 Introduction	134
13.2.5.2. MIME et le monde de l'EDI	136
13.2.6. Internet en Belgique: Belnet	137
13.3. Le réseau des facultés	137
13.3.1. Les échanges entre Eudora et Leibniz	138

CHAPITRE 14. SYNTHÈSE ET CONCLUSION

140

BIBLIOGRAPHIE

142

Abstract:

This work is intended to assess the implementation of an EDI relationship between the University of Namur and the Monash University of Curtin (Australia). This assessment deals primarily with the methodological and technical aspects characterising the EDI implementation.

A general analysis of the main choices that arise when somebody wants to undertake an EDI relationship comes after a general description about what we call « the EDI phenomenon ». Later, we discuss the nature and the importance of the features that we must consider in order to assess properly the quality of an EDI software. Based on these conclusions, we undertake a detailed description of the EDI software we use: ATLAS EDI.

Our work does not consist of only performing a theoretical study of the EDI matter but takes its originality from the fact that we describe for the first time the technical implementation of an EDI relationship. This work involves the detailed description of the best way to make a good configuration of the software.

Other related topics are also depicted: a comparison with the functionalities offered by an other type of EDI translator (the Mapper), the methodological point of view of the design of EDIFACT messages, the design of an interface in a Windows environment and the analysis of the way messages are transmitted between us and our partner in Australia.

Introduction

Ce travail est consacré à l'évaluation de la mise en oeuvre d'une relation EDI entre l'institut d'Informatique de Namur et la Monash University de Curtin (Australie). Cette évaluation porte essentiellement sur les aspects méthodologiques et techniques de cette mise en oeuvre.

Après une présentation globale du phénomène EDI, nous proposons une analyse générale des choix à effectuer pour mettre en oeuvre des échanges EDI. Ensuite, nous abordons les critères de choix destinés à classer et à évaluer les logiciels EDI. Sur base de ces critères, nous présentons les propriétés du logiciel ATLAS EDI.

Notre travail ne se limite pas à une simple étude théorique sur la mise en oeuvre d'échanges EDI. En effet, sur base de l'expérience EDI des équipes universitaires de Curtin et Maribor (Slovénie), nous avons réalisé une première expérience EDI concrète au sein de l'Institut d'Informatique. Sur base d'échanges avec notre collaborateur australien Craig Parker, nous avons pu évaluer concrètement comment mettre sur pied une relation EDI. Nous avons décrit l'ensemble des opérations à effectuer pour configurer ce type de logiciel. Nous avons également décrit les opérations nécessaires pour réaliser des traductions de fichiers EDIFACT à l'aide de ce logiciel. Pour mieux cerner les possibilités de ce logiciel, nous avons aussi réalisé une comparaison de ses fonctionnalités avec celles offertes par un autre logiciel de traduction EDI, le Mapper d'Interbridge.

Afin d'illustrer la possibilité d'interfacer ATLAS EDI avec d'autres applications, nous présentons également une interface de manipulation de fichiers plats définie pour un type de message EDIFACT.

L'analyse technique des moyens de communication (le mail électronique) utilisés pour réaliser les échanges EDI permet de comprendre comment les échanges de messages s'effectuent. Cette analyse pourrait constituer la base de la réalisation d'une connexion automatique entre ATLAS EDI et le réseau Internet. Elle permet également d'introduire la notion d'encapsulation de messages EDI dans du courrier électronique.

Chapitre 1. Le phénomène EDI

1.1. Naissance du concept d'EDI

Le développement explosif des technologies, particulièrement des télécommunications, a déjà conduit à de nombreux changements dans l'organisation des sociétés qui prennent part à cette révolution. Aucune branche d'activité n'échappe plus désormais à ce tourbillon technologique qui s'est popularisé au point de pénétrer au sein de la plupart des foyers.

Au centre de toutes les convoitises se trouve l'information. L'information est devenue aussi importante pour une organisation que ses employés, ses équipements et ses bâtiments. La structuration de l'information conditionne sa capacité à être traitée. Les traitements sont aussi divers que les contextes dans lesquels ils prennent place. Il peut s'agir de vérifier des faits, d'analyser des données, d'en faire la synthèse et de supporter un processus de décision. En toute généralité, la valeur d'une information est directement dépendante de l'entité (personne ou programme, par exemple) qui peut y accéder [ROSE, 1990]. En effet, la valeur d'une information donnée peut augmenter simplement en la déplaçant vers un endroit où une entité peut en faire usage. C'est donc la mobilité de l'information qui lui confère sa valeur. Les entreprises ont dès lors tout intérêt à optimiser la structure et le transfert des informations dont elles dépendent pour mener leurs activités, d'où le rôle crucial joué par les réseaux de communication dans le transfert de données.

Tous les ingrédients sont ainsi présents pour construire une nouvelle façon d'établir des contacts commerciaux tirant profit des potentialités offertes par les technologies informatiques. Tout naturellement, on s'est livré à des échanges de données informatisés, c'est à dire à de l'EDI (Electronic Data Interchange), afin de remplacer les procédures traditionnelles d'échange de documents sur papier. Ainsi, les bons de commandes, les factures et autres accusés de réception ont pris la forme de fichiers de données à échanger par le biais de réseaux informatiques.

1.2. Définition de l'EDI

Comme nous venons de l'expliquer, l'Electronic Data Interchange trouve son origine dans le contexte du monde des affaires. D'un point de vue informatique, il peut être défini [GEVERS, 1993] comme un « échange de données électroniques réalisé entre applications (ou bases de données) informatiques conçues indépendamment et de même niveau et effectué sur base de spécifications standardisées (ou convenues bilatéralement) de la représentation et généralement aussi de la structuration et surtout de la sémantique des données échangées ». L'auteur insiste sur la nécessité de différencier les propriétés de l'EDI par rapport à des techniques informatiques voisines, ce qui justifie l'inhabituelle complexité de la définition. Aussi, nous allons commenter les termes soulignés afin de préciser une fois pour toutes ce que l'on entend précisément par EDI.

L'échange doit avoir lieu entre applications. Cela signifie que les données échangées dans le cadre d'une relation commerciale entre plusieurs partenaires sont issues de traitements spécifiques de l'information élaborés par chaque protagoniste. En fait, chaque organisation communicante possède son propre système d'informations et gère comme elle l'entend ses bases de données. L'EDI établit de la sorte un échange entre des données organisées indépendamment les unes des autres. Il est important de bien différencier cette démarche de celle qui consiste à mettre en oeuvre simplement une messagerie de type courrier électronique entre personnes.

Les applications impliquées dans l'EDI sont « de même niveau ». En clair, l'EDI se différencie des opérations de consultation et de mise à jour de bases de données éloignées par une application. Dans ce cas, l'application a recours aux services d'un SGBD qui n'est donc pas au même niveau puisqu'il est utilisé par l'application (modèle client-serveur).

Une simple connexion physique entre applications ne suffit pas pour mettre en oeuvre l'EDI. Pour qu'il y ait communication, il faut que les partenaires suivent des normes quant à la façon de réaliser les échanges et qu'ils partagent une représentation commune du contenu des échanges. La communication et la représentation des données doivent idéalement être standardisées afin de réduire la variété, et donc le risque d'incompatibilité, de leur mise en oeuvre. On parlera donc de standards de communication (pour la réalisation des échanges au niveau informatique) et de standards de représentation (relatifs au contenu des échanges). Soulignons le fait que les standards de communication relèvent typiquement du monde des informaticiens alors que l'évolution des standards de représentation est liée à leurs utilisateurs, à savoir les acteurs économiques qui se livrent aux échanges EDI.

Les données échangées au moyen de l'EDI sont caractérisées par les critères suivants, adoptés par tous les partenaires :

- un certain type de représentation (mode d'identification, taille, type de caractères,...);
- un mode particulier de structuration (disposition relative des données à échanger);
- une sémantique bien définie (chacune des données possède une signification bien précise).

Ce dernier aspect singularise l'EDI par rapport aux techniques de transfert de fichiers de type FTAM qui se déroulent également entre systèmes informatiques indépendants. Dans ce cas, aucune sémantique particulière n'est associée aux fichiers échangés.

Ajoutons à cette définition générale le fait que les fonctionnalités de l'EDI peuvent revêtir plusieurs aspects selon la nature du contenu des échanges. Dans [OSITOP, 1991, p5] une classification généralement admise est présentée sous la forme des trois entités suivantes:

-le « transaction data interchange » subdivisé lui-même en:

-« general data interchange »: il s'agit des transactions commerciales générales qui concernent les transactions d'affaires ou administratives telles que les factures, les bons de commandes, les instructions de paiement, les notes de crédit, ...

-« specific data interchange »: ce sont les transactions commerciales spécifiques (élaborées à des fins spéciales ou particulières) telles que le transfert de fonds électronique (utilisé largement par le monde bancaire).

-le « technical data interchange », ou échange de données techniques, utilisé notamment pour échanger des plans et des dessins et pour la conception et la mise en oeuvre de projets tels que le Computer Aided Design/Computer Aided Manufacturing (CAD/CAM).

-l'EDI interactif, utilisé par exemple pour la réservation de billets d'avion.

Dans la littérature, le terme EDI désigne presque toujours les transactions commerciales générales pour lesquelles les Nations unies mettent au point un standard de représentation, UN/EDIFACT, que nous étudierons plus loin.

1.3. Nécessité de standards pour la mise en oeuvre de l'EDI

Comme nous l'avons déjà signalé, il est commode de recourir à des standards à la fois pour la communication proprement dite et pour la représentation des données échangées. Dans le cadre de notre propos, le terme « standard » équivaut à « norme ».

1.3.1. Niveaux de standardisation

Par définition, l'EDI se déroule entre deux applications implantées sur des systèmes informatiques distincts et indépendants reliés au moyen d'un réseau de télécommunication. La figure 1.1 représente schématiquement deux entités communicantes sur lesquelles se trouvent respectivement les applications A et B. Le schéma suggère l'existence de différentes couches, au sens traditionnel, chacune étant responsable d'une fonction particulière intervenant dans la migration des données.

Ce modèle comporte une simplification de l'architecture en couches conforme à OSI ou au monde TCP/IP. C'est d'ailleurs dans ce contexte qu'il faut interpréter les transferts de données entre les deux systèmes.

Afin de comprendre le principe de l'EDI, il est bon de définir un ensemble hiérarchique de trois couches fonctionnelles [GEVERS, 1993] pour chaque site.

La couche supérieure est celle où est implantée l'application de l'utilisateur. Celui-ci possède sa propre représentation des données relatives aux concepts cibles des échanges (facture, bon de commande,...).

La couche moyenne est responsable de la traduction dans un formalisme connu du partenaire des données de l'utilisateur exprimées selon ses propres conventions. Ce formalisme commun permet aux différents partenaires de se communiquer sans aucune ambiguïté des données que chacun exprime de façon indépendante des autres. C'est à ce niveau qu'intervient la notion de standard de représentation de données.

La couche inférieure est chargée de la communication proprement dite. Les différents aspects de la communication (accès au réseau, opérations de transfert, ...) sont organisés selon des normes bien précises, appelées dans notre propos des

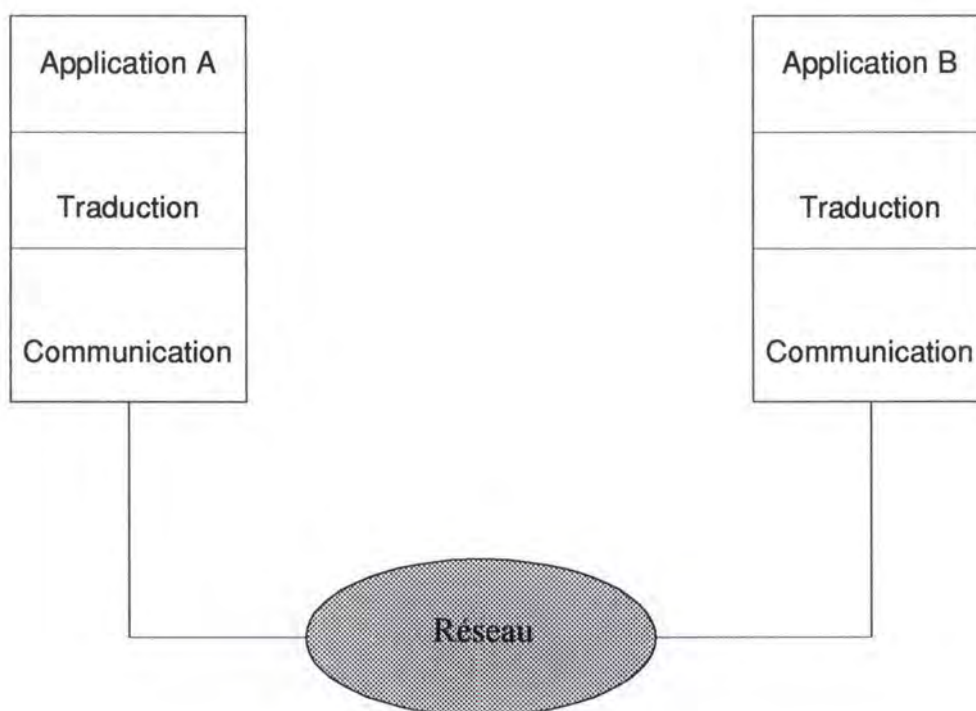


Figure 1.1: Echange entre applications distantes (d'après [GEVERS, 1993])

standards de communication.

Dans ce qui suit, nous allons résumer les propriétés principales des standards de représentation et de communication, notions omniprésentes dans notre étude.

1.3.2. Avantage des standards de représentation des données

Les utilisateurs de l'EDI sont contraints d'exprimer les données qu'ils échangent sous une forme telle que leurs partenaires puissent les interpréter et les traiter sans ambiguïté. Cette nécessité n'apparaît qu'au moment où des données doivent être échangées. En effet, chaque utilisateur organise et représente les données de son organisation (système d'information) comme il l'entend et de façon indépendante de ses partenaires potentiels. Un standard de représentation donné désigne un langage commun à un ensemble de partenaires utilisé pour exprimer les messages qu'ils échangent. Pour J.Berge [BERGE, 1991, p13], un standard est « la représentation admise de l'information (quelles données et la façon dont elles sont structurées et assemblées) à transmettre d'une application informatique à une autre ».

L'échange de données peut concerner seulement deux partenaires tout comme il peut en impliquer un nombre a priori illimité. Dans le premier cas, un simple consensus entre les deux parties, soient P1 et P2, sur le mode de représentation des données suffit. Cependant, si l'un des deux protagonistes, par exemple P2, décide de réaliser des échanges avec une entité P3 (supposée ne pas encore s'adonner à l'EDI), alors se pose la question du choix du mode de représentation:

- P2 adopte celui proposé par P3, ce qui l'oblige à gérer deux modes (un pour P1 et un pour P3);
- ou bien P3 se plie aux conventions passées entre P1 et P2, ce qui lui permet aussi de se lancer dans des échanges avec P1 s'il le désire, sans devoir renégocier un nouveau système avec lui.

Nous voyons que la complexité de ce problème augmente très fort si le nombre de liens entre les partenaires est élevé. La situation devient impossible à gérer par cette méthode de conventions bilatérales. En effet, si n entités pratiquent de l'EDI de cette façon, et que chacune établit un lien avec les $(n-1)$ autres, alors le nombre N de conventions bilatérales à conclure est:

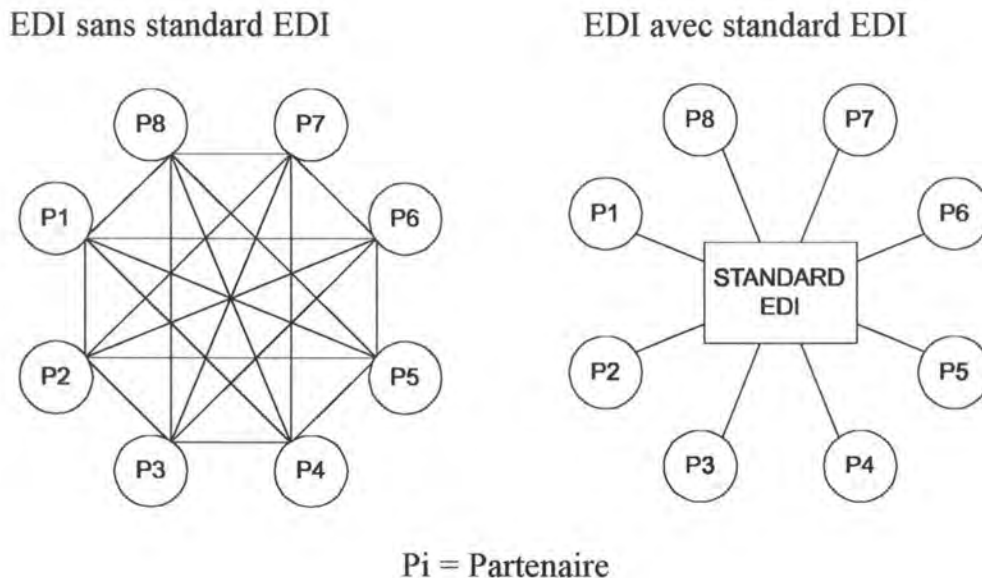
$$N=n(n-1)/2$$

Bien entendu, de telles conventions ne doivent pas être considérées comme étant des « standards ». La notion de standard prend son sens lorsqu'un nombre significatif d'utilisateurs l'adoptent. La figure 1.2 illustre notre propos à l'aide d'un schéma qui représente les rapports extrêmes que peuvent établir huit partenaires:

- la partie gauche représente le nombre maximal (28) de conventions bilatérales passées entre les différents partenaires;

-la partie droite montre que les mêmes relations sont nettement simplifiées par l'adoption d'un standard.

L'avantage d'un standard commun se traduit notamment par une réduction des coûts liés aux échanges. De plus, dans cet exemple, si un partenaire respectueux du standard se joint aux huit autres, ce sont huit nouvelles relations bilatérales potentielles qui sont créées d'un seul coup.



**Figure 1.2: L'EDI avec et sans standard
(d'après [GEVERS, 1993])**

1.3.3. Standards de communication

L'EDI doit son existence aux services offerts par les réseaux de télécommunication qui servent de support au transfert des messages. Le monde des réseaux est peuplé de très nombreux représentants et il s'avère nécessaire de définir quelques standards afin de clarifier les choses, surtout lorsque les communications s'étendent le long d'un périmètre international.

Dans sa classification des standards de communication [GEVERS, 1993], l'auteur établit la distinction entre les standards propriétaires, *de facto* et *de jure*.

Les protocoles mis au point par IBM, par exemple, sont largement représentés dans la première catégorie.

Les standards *de facto* comptent dans leurs rangs l'OFTP (ODETTE File Transfer Protocol), et l'ensemble de protocoles TCP/IP.

C'est probablement X.400 qui (standard *de jure*) présente le plus grand intérêt pour l'EDI et surtout le protocole X.435, issu de X.400, spécialement conçu pour l'EDI.

Nous ne nous étendrons pas plus ici sur la présentation des standards, nous renvoyons le lecteur curieux à [GEVERS, 1993] et à [BERGE, 1991].

1.4. Présentation générale du standard UN/EDIFACT

Le standard UN/EDIFACT (United Nations/Electronic Data Interchange for Administration, Commerce and Transport) a été conçu par les Nations unies (plus précisément par l'UNECE, United Nations Economic Commission for Europe) afin de développer le recours à l'EDI en Europe.

Un message EDIFACT est un ensemble d'informations à envoyer afin d'accomplir une fonction dans le cadre d'un échange commercial. Par rapport à un document sur papier, un message EDIFACT est aussi composé de sections, à leur tour composées de libellés. En fait, on peut établir [BERGE, 1991] une correspondance entre un message EDIFACT donné et le formulaire papier classique qui a la même fonction:

- dans un texte « normal », nous avons les entités mots, phrases, paragraphes/chapitres et grammaire;
- ces entités ont leur équivalent respectif en EDIFACT: les éléments de données, les segments, les messages et la syntaxe.

Ainsi, les éléments de données EDI (mots) sont rassemblés dans des segments (phrases) qui sont groupés en messages (paragraphes/chapitres), le tout conformément à une syntaxe (grammaire).

La figure 1.3 illustre la correspondance entre la représentation sur papier et en EDIFACT.

EDIFACT	Comparaison avec un formulaire	Représentation dans le standard
Elément de donnée	donnée écrite	valeur pour la donnée
Segment	contenu d'une rubrique	groupe logique et identifié
Message	formulaire complété	collection de valeurs relatives à une fonction commerciale

Figure 1.3: Table de correspondance entre représentation sur papier et en EDIFACT

Le standard EDIFACT est structuré hiérarchiquement en entités que l'on peut mettre en relation avec les concepts qui décrivent les documents usuels rédigés en langage naturel.

Ces définitions nous permettent de mieux préciser ce que recouvre le terme EDIFACT. En fait, de nombreux documents sont concernés, dont les suivants:

- les règles de syntaxe EDIFACT (ISO 9735);
- le répertoire des éléments de données EDIFACT;
- un répertoire des segments;
- un répertoire des messages standards des Nations unies.

Chaque élément de donnée est décrit dans le répertoire par:

- un identifiant (ou code);
- un titre;
- une description;
- un format (longueur et type des données);
- une note (observation supplémentaire).

Nous verrons que chaque élément de donnée peut être lui-même composé d'autres éléments. Nous parlerons alors d'éléments de données composites constitués d'éléments de données simples.

Nous présenterons en détail des messages EDIFACT plus loin dans notre étude. Nous discuterons également de leur mise en oeuvre pour réaliser des échanges EDI.

Notons dès à présent qu'il existe plusieurs évolutions du standard EDIFACT. A chaque évolution sont associées de nouvelles versions des répertoires (répertoires des éléments de données, des segments et des messages). Lorsque l'on veut préciser l'édition d'EDIFACT que l'on utilise, on doit donc mentionner l'édition des répertoires utilisés (90.1 ou 92.1 par exemple).

1.5. Autres types de standards de représentation

La littérature sur les standards présente une image très complexe de leurs différents modes de classification. Dans ce domaine, il est bon de disposer de repères afin de les discerner. Aussi, nous conseillons au lecteur avide de clarté de lire une synthèse sur le sujet parue dans [GEVERS, 1994].

Nous donnons ici quelques éléments de comparaison et de différenciation avec EDIFACT succinctement présenté ci-dessus.

Il faut savoir que plusieurs versions d'un même standard peuvent coexister (cf. EDIFACT) et qu'ils évoluent avec le temps.

Une même syntaxe peut être utilisée afin d'élaborer différents standards EDI. Inversement, un standard EDI peut utiliser différentes syntaxes pour constituer ses versions consécutives.

Par ailleurs, tous les standards ne sont pas structurés aussi nettement qu'EDIFACT. Certains standards EDI peuvent ne pas être divisés en entités séparées comme c'est le cas pour EDIFACT. Par exemple, ils peuvent n'offrir qu'un type de répertoire de messages accompagné de beaucoup d'explications. Dans ce cas, la syntaxe n'est pas explicitée sous forme de règles contenues dans une partie isolée, mais sont incluses dans le répertoire même.

La littérature décrit presque toujours les messages EDI comme des objets structurés hiérarchiquement comme le sont les documents commerciaux en papier. Or, certains standards ne possèdent pas cette structure hiérarchique.

Il est à noter que l'inclusion des répertoires dans la définition d'un standard distingue les standards EDI des autres standards de représentation utilisés pour la communication entre ordinateurs (ex: l'Abstract Syntax Notation, ASN.1 de l'OSI)

Généralement, les standards sont aussi classés selon le contexte dans lequel ils ont été conçus et selon la manière dont leur développement est influencé. Ainsi, on parle de standards propriétaire, *de facto* et *de jure*.

Le terme propriétaire désigne un standard conçu par une entité pour l'ensemble de ses échanges avec ceux qui adopteront le même standard.

Selon Ositop [OSITOP, 1991, p17], les standards *de facto* naissent d'un « regroupement d'utilisateurs qui identifient un besoin commun et produisent un standard de façon coopérative pour le satisfaire ». Les standards bilatéraux, à cause de leur nature spontanée et volontaire, pourraient aussi être considérés comme des standards *de facto*.

Par définition, les standards *de jure* sont développés par une autorité reconnue. Ce genre de développement implique la rencontre d'un large consensus. Donc, les standards correspondants ont besoin de plus de temps pour être développés que les autres mais devraient être plus stables à long terme. EDIFACT est un exemple typique de standard *de jure*.

1.6. Les contextes d'intégration de l'EDI

Considérons une organisation désireuse d'implémenter un système EDI. Pour ce faire, elle doit notamment définir le contexte dans lequel elle veut entreprendre les échanges. Dès lors, un certain nombre de choix sont à réaliser en ce qui concerne

les partenaires commerciaux, les types de messages à échanger, les standards utilisés, ...etc.

Parmi les différentes fonctionnalités caractéristiques de l'implémentation de l'EDI, il en est une qui est évidemment fondamentale. Elle concerne la traduction des données traitées par l'utilisateur en données exprimées selon un formalisme propre à un standard de représentation EDI. En effet, comme chaque utilisateur possède ses propres représentations de données, il est nécessaire de les exprimer sous une forme unique, reconnue par ses partenaires. La conversion des données du format utilisateur vers le format EDI (et dans l'autre sens) est effectuée par des logiciels de traduction EDI. Nous détaillerons ce processus dans la suite de cette étude.

Autour de cette fonctionnalité de base viennent se greffer des architectures fort différentes selon les cas. En effet, il existe de nombreuses implémentations possibles de l'EDI en fonction du degré d'informatisation du traitement interne des données choisi par l'utilisateur.

Nous allons dépeindre les différents choix qui se présentent une fois qu'un utilisateur a pris la décision d'implémenter l'EDI dans un environnement caractérisé par un système d'information particulier. Ce choix définit la profondeur de l'intégration de l'EDI.

Gerson [GERSON, 1991] définit cinq scénarios d'intégration de l'EDI:

- le système sur papier;
- l'intégration à la base de données existante;
- l'intégration aux processus et aux applications existants;
- le développement de nouvelles applications et de nouveaux processus basés sur l'EDI;
- le développement d'un système application/EDI complètement intégré.

1.6.1. Le système papier

Le principe d'un système EDI papier au sein d'une organisation est illustré à la figure 1.3.

Dans le cas de la réception d'informations EDI, ce système n'assure aucun transfert de données au-delà du processus de traduction. Dans ce cas, les données reçues par EDI sont traduites et stockées dans un fichier intermédiaire. Ces données peuvent être imprimées pour un usage ultérieur quelconque et subir un traitement totalement manuel.

Dans le cas de l'émission de données EDI, les données sont fournies manuellement au traducteur EDI au moyen d'un clavier classique. L'encodage des données, pouvant être assisté par le traducteur (par des interfaces de saisie de données), résulte en la création d'un fichier intermédiaire traitable par le logiciel et prêt à être traduit en format EDI.

Dans ces deux cas, le fichier intermédiaire est appelé fichier plat et désigne:

- soit, en cas de réception, le fichier de données qui après réception a été traduit du format EDI en format propre à l'utilisateur;
- soit, en cas d'émission, le fichier des données destiné à être traduit (dans un format EDI) puis transmis.

Nous reviendrons abondamment sur la notion de fichier plat et de fichier EDI (EDIFACT) dans la suite. Néanmoins, il est bon de fixer déjà les idées à propos du principe de la traduction d'un fichier plat en un fichier EDI et vice-versa. Le principe général de la traduction repose sur l'existence d'une table de traduction (mapping table) qui décrit l'emplacement des différents éléments de données dans le fichier plat et dans le fichier EDI.

Dans le cas d'une traduction du fichier plat vers un fichier EDI, la table renseigne au traducteur la position dans le fichier plat des éléments de données à extraire et à intégrer dans le fichier EDI à transmettre. Notons que, pour un type de message donné et une version précise du standard utilisé, la position des éléments de données dans le fichier EDI est rigoureusement fixe. Au contraire, l'utilisateur définit des fichiers plats dont les éléments de données respectent un ordre qui lui convient le mieux. Du côté récepteur, la table indique la position que doivent prendre les éléments de données dans le fichier plat de l'utilisateur. Bien entendu, les utilisateurs émetteur et récepteur doivent chacun définir leur propre table de conversion pour chaque type de message, ou, ce qui revient au même, pour chaque type de fichier plat correspondant à un type de message.

L'auteur fait remarquer que ces systèmes papier sont le plus souvent utilisés par de petites sociétés qui ne sont pas (encore) prêtes à intégrer complètement l'EDI dans leur système informatique. Souvent, ces sociétés sont amenées à recourir à l'EDI sous la pression d'un partenaire commercial important qui, lui, fait un usage courant de ce moyen d'échange. Dans ces conditions, elles n'ont pas nécessairement intérêt à développer un système complètement intégré.

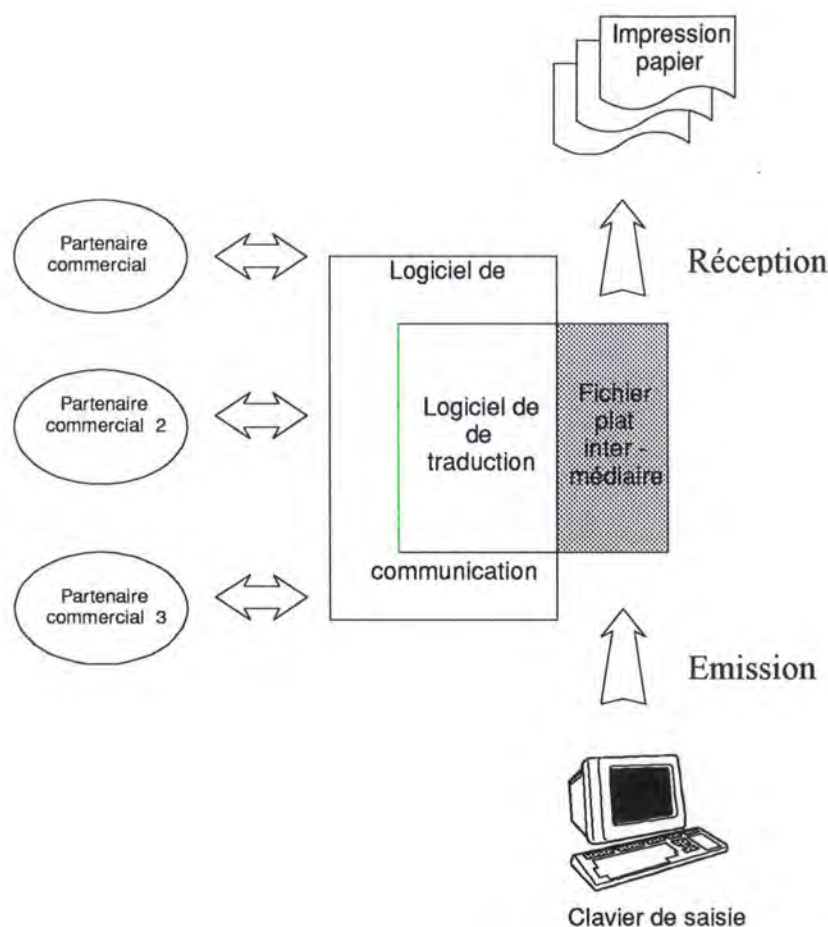


Figure 1.3: Le système EDI sur papier au sein d'une organisation.

D'autres sociétés, plus grandes, peuvent aussi adopter ce système élémentaire pour estimer l'apport que représenterait pour elles le recours intensif à l'EDI. Ces développements pilotes peuvent servir d'amorce à une intégration ultérieure.

1.6.2. Intégration avec la base de données existante de l'organisation

L'organisation peut très bien établir un lien physique entre les données contenues dans sa base de données et le traducteur EDI qu'elle utilise pour ses échanges. Ce mode d'intégration revient en quelque sorte à interfacer la base de données avec le logiciel de traduction.

Par rapport au système précédent, il existe toujours un fichier plat qui sert d'« input » ou d'« output » au logiciel de traduction. La principale caractéristique de ce mode d'intégration est l'existence d'un lien entre les données du fichier plat et celles de la base de données de l'utilisateur. Ce lien est concrétisé par l'intervention d'un programme qui assure l'injection de données du fichier plat vers la base de données et vice-versa. Bien entendu, ces transferts ne sont destinés qu'à relocaliser

des données et, dans ce mode d'intégration, il n'y a aucun processus activé par ces données. En effet, dès qu'elles parviennent à l'utilisateur au terme d'un transfert EDI, elles sont simplement stockées dans la base de données.

La figure 1.4 illustre ce processus général de « data mapping » dans le cadre de l'intégration décrite ici. Dans ce cas, l'auteur parle plutôt de « database mapping » puisqu'il ne s'agit, à ce stade, que d'un processus de mapping de données. Ces données ne sont par définition pas utilisées dans l'immédiat mais sont stockées pour un usage futur. « Le terme plus générique "data mapping" est utilisé pour inclure l'intégration d'applications, l'usage immédiat des valeurs faisant partie d'un processus ou d'un appel de procédure où les valeurs reçues sont des paramètres. » [GERSON, 1991, p110].

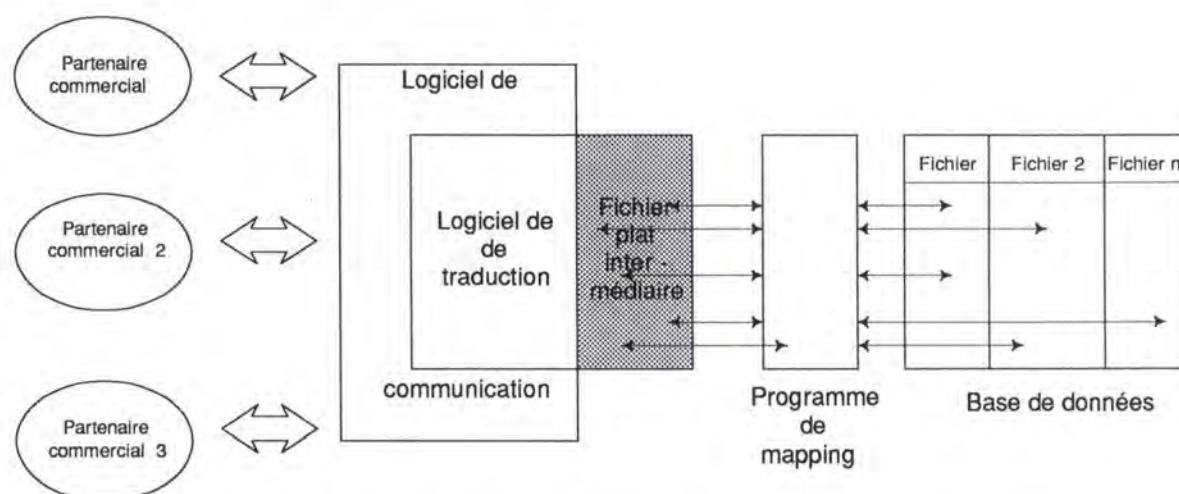


Figure 1.4: Intégration de l'EDI à une base de données existante.

Dans ce type de système, des fonctionnalités de stockage et d'extraction des données se font à l'aide d'un programme de mapping. Notons qu'il ne s'agit que d'un transfert de données entre fichiers.

1.6.3. Intégration à des applications et à des processus existants

Des processus de traitement spécifique des données échangées par EDI peuvent être implémentés par les utilisateurs. Par rapport au « database mapping » ce scénario implique une extraction des données du fichier plat à passer à des procédures qui forment l'application de l'utilisateur. Ce n'est donc plus un simple mapping de valeurs entre un fichier plat et des fichiers de la base de données (et vice-versa).

Les données du fichier plat sont des paramètres à passer à des procédures qui se chargent de les replacer dans la base de données et vice-versa.

La figure 1.5 représente le principe de ce type d'intégration.

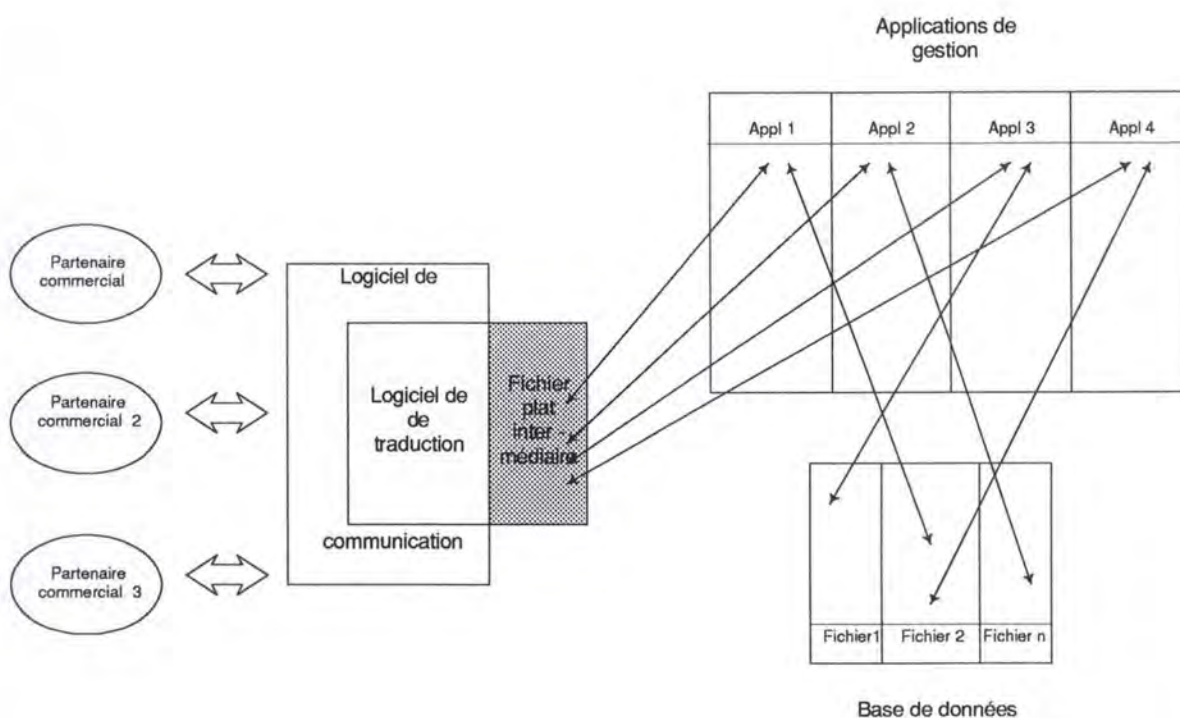


Figure 1.5: Intégration de l'EDI à des applications existantes.

Un exemple de ce type d'intégration serait l'incorporation de l'EDI dans un système existant qui permettrait l'exécution informatisée d'ordres d'achat. Ainsi, des ordres d'achat seraient automatiquement exécutés dès la réception de messages EDI correspondants.

1.6.4. Développement de nouvelles applications et de nouveaux processus basés sur l'EDI

Au lieu de se limiter à utiliser l'EDI pour effectuer des processus qui existaient déjà avant son intégration, l'utilisateur peut en profiter pour développer de nouvelles applications.

Cette méthode implique donc des aménagements plus importants: nouvelles fonctionnalités, nouveaux types de messages,... Ces transformations sont bien sûr destinées à être à leur tour intégrées dans le système d'information global.

L'illustration de ce scénario est représentée à la figure 1.6.

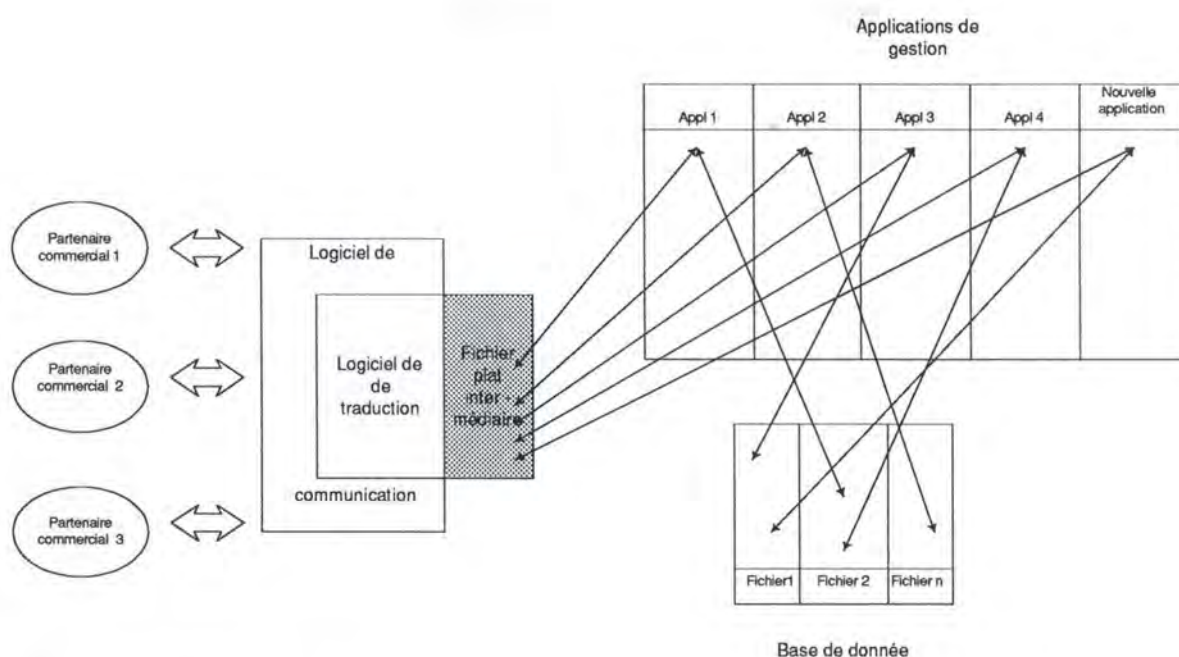


Figure 1.6: Développement de nouveaux processus tirant profit des potentiels offerts par l'EDI.

1.6.5. Développement d'un système qui intègre complètement les applications à l'EDI

Nous arrivons ici au niveau d'intégration le plus profond puisqu'il s'agit d'inclure l'intégration EDI dans la conception d'un nouveau système de traitement de données commerciales.

La figure 1.7 dépeint cette situation d'intégration complète caractérisée par un couplage étroit entre les modules du système et le logiciel de traduction. Remarquons qu'il n'est plus besoin d'un fichier plat intermédiaire puisque, par définition, l'intégration est totale entre les modules d'application et les modules du logiciel de traduction.

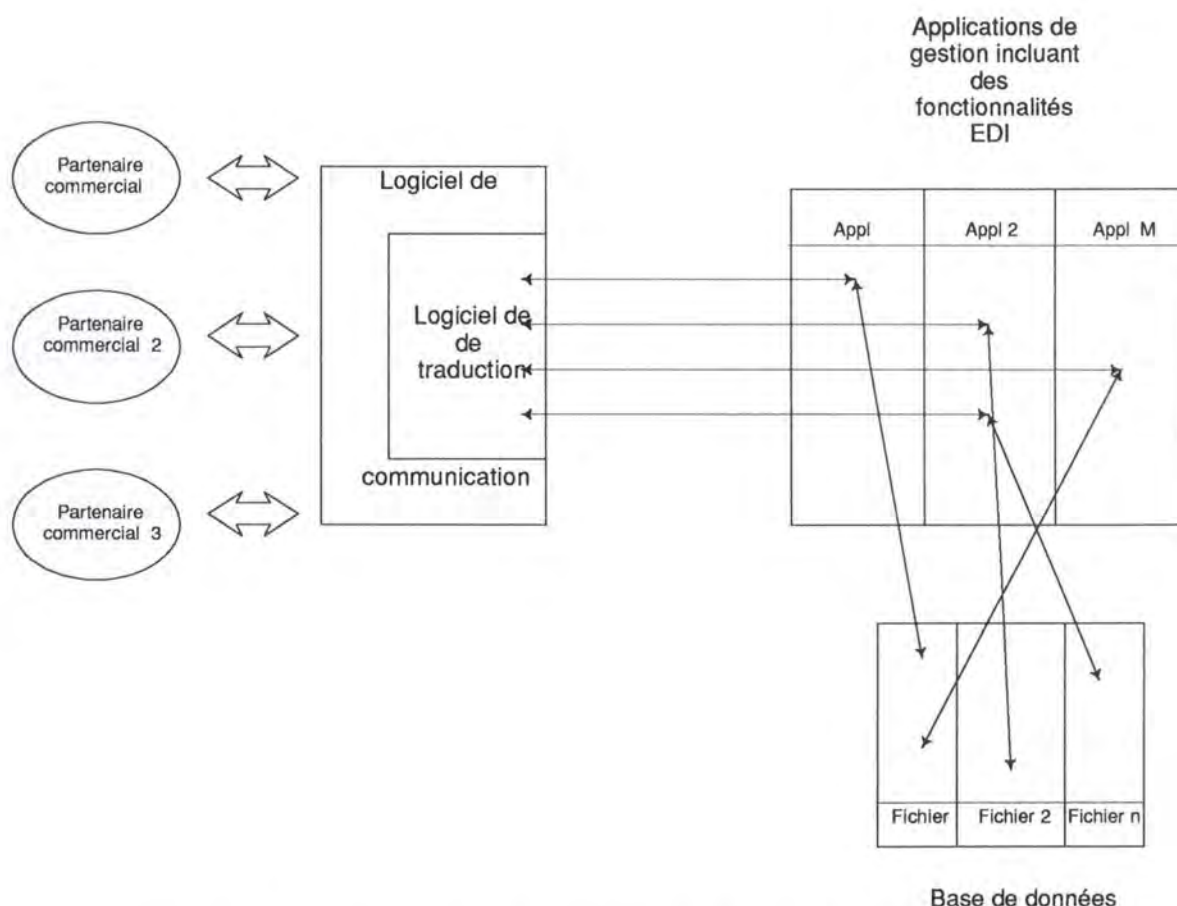


Figure 1.7: Intégration complète de l'EDI aux applications.

1.6.6. Conclusion et remarques

Les différents types d'intégration de l'EDI présentés dans [GERSON, 1991] correspondent à des situations réelles et peuvent aider les utilisateurs potentiels à se faire une première idée de la façon dont ils envisagent le recours à l'EDI dans le cadre de leurs activités.

Aussi, il est intéressant de formuler quelques commentaires et remarques à propos de ce qui vient d'être décrit.

Par définition, le mapping entre les données traduites et les données contenues dans la base de données d'une organisation n'existe pas dans un système où la source des valeurs est une feuille de papier. Selon [GERSON, 1991] la plupart des logiciels de traduction fournissent des écrans prédéfinis ou définis par l'utilisateur afin d'assister le processus d'encodage. Nous reviendrons largement sur ce point dans la suite de notre étude. Ce qui est automatique, c'est la traduction des valeurs d'entrée et leur communication au partenaire commercial. Chez ce partenaire, les informations reçues sont traitées selon des procédures qui lui sont propres. Comme tous les logiciels de traduction disponibles pour des micro-ordinateurs incluent des

fonctionnalités de saisie de données et d'impression, l'utilisateur peut se contenter dans ce cas élémentaire d'une connaissance rudimentaire du fonctionnement interne du traducteur. De plus, aucune compétence en programmation n'est indispensable pour ce genre d'intégration.

Dans le deuxième cas d'intégration que nous avons vu, les valeurs des données communiquées aux utilisateurs sont stockées dans la base de données et ne sont pas nécessairement traitées immédiatement. Le processus inverse, l'extraction de valeurs de la base de données pour leur insertion dans un échange EDI existe également. Le stockage des données reçues implique leur traduction, leur validation et leur injection dans des emplacements prédéfinis de la base de données. Ces endroits dépendent des types d'éléments de données et des messages dans lesquels ils sont contenus. Par conséquent, l'utilisateur doit être capable d'analyser les composants des messages qu'il échange. Cela implique de maîtriser à la fois la définition des messages et la documentation sur les fichiers plats fournie par le vendeur. Notons que, dans certains cas, il est nécessaire de transformer des données reçues quand leur format diffère du format défini par l'utilisateur émetteur. Ce peuvent être des conversions mineures de valeurs telles que le changement de l'« heure EDI » exprimée selon une horloge de 24 heures vers le mode AM/PM et la transformation de la date EDI standard (YYMMDD) en une autre représentation.

Il est clair que la complexité qui apparaît dès ce deuxième mode d'intégration doit être prise en charge au moyen de la programmation de différents processus: le processus de mapping, la définition d'événements déclencheurs du mapping (une information extérieure, le temps, la présence de valeurs dans le fichier intermédiaire), le traitement des erreurs (syntaxiques et sémantiques),...

Cette étape de programmation s'ajoute à celle qui consiste à définir les tables de conversion entre les fichiers plats et EDI.

Le troisième scénario d'intégration de l'EDI est caractérisé par la génération d'actions spécifiques causées par la réception d'un message EDI (plutôt que leur simple stockage). Par exemple, la réception d'un bon de commande EDI peut résulter en un traitement informatisé comme il le serait s'il était reçu par téléphone et encodé manuellement. Au lieu d'un stockage de valeurs dans une base de données, le processus de mapping effectue des appels de procédures spécifiques. Ce scénario implique que l'analyste utilisateur prenne en compte les éléments suivants:

- la procédure interne à appeler lorsqu'un type de message EDI survient;
- le rôle joué par les éléments du message en tant que paramètres d'appel de procédures;
- l'ajustement des données reçues aux exigences syntaxiques que doivent satisfaire les valeurs stockées dans les tables de la base de données;

Dans le quatrième type d'intégration, on assiste au développement de nouveaux processus internes à l'organisation. Les activités incluent donc à la fois le mapping

de la base de données et l'intégration d'applications. De même le nouveau système logiciel doit être conçu en tenant compte des caractéristiques suivantes:

- le format désiré par l'utilisateur;
- les fonctionnalités que l'utilisateur a l'intention d'implémenter;
- les capacités du système de gestion de la base de données.

Enfin, le développement d'un système qui intègre complètement les applications et l'EDI implique qu'aucun système capable de fournir des données à partir de données reçues n'existe avant l'intervention d'un développement spécifique. L'utilisateur développeur doit faire face au développement d'un système entièrement neuf selon différents types d'approches possibles qui vont de l'implémentation d'un système entier (le traducteur y compris) à l'adaptation des applications à un logiciel de traduction standard.

Chapitre 2. Portée du travail

Après ce rapide récapitulatif des principales caractéristiques de l'EDI importantes pour la compréhension des prochains développements, nous décrivons à présent le contexte dans lequel s'inscrit notre étude. Ensuite, nous définissons nos objectifs sous un angle technique, en référence aux éléments présentés dans la première partie.

2.1. Contexte et objectifs généraux

Ce travail s'inspire d'un projet présenté au mois de juin 1994 au sommet de Bled (Slovénie) par deux équipes de chercheurs universitaires passionnés par la diffusion de l'EDI [PARKER, 1994]. L'une travaille à l'Université de Maribor, en Slovénie, et l'autre mène ses recherches dans une université australienne (Monash University). Comme nous allons le voir, elles ont supervisé, chacune dans leur institution respective, la mise en pratique par des étudiants d'échanges commerciaux entre les deux pays au moyen de l'EDI. Cette simulation d'échanges commerciaux nous a servi de référence pour tester la mise en oeuvre d'un système EDI dans notre Institut d'Informatique.

2.1.1. Simulation d'échanges EDI Curtin/Maribor

Dans leur article [PARKER, 1994], les auteurs qualifient leurs recherches d'« étude pilote d'une simulation réalisée dans des contextes pédagogiques variés incluant la gestion des réseaux, la conception des systèmes d'information et le marketing ».

La simulation a été menée conjointement par l'université de technologie de Curtin (Australie) et l'université de Maribor (Slovénie). Les étudiants de Curtin devaient se familiariser avec les principes des télécommunications appliquées dans un environnement international tandis que ceux de Maribor avaient comme objectif l'apprentissage du commerce électronique.

Les résultats de cette étude suggèrent qu'il est techniquement possible d'implémenter une simulation entre universités distantes. C'est aussi une illustration de la possibilité d'établir des échanges commerciaux au sein de la communauté internationale.

Il n'est pas inutile d'insister sur la mise en oeuvre de l'EDI, même aujourd'hui. En effet, « beaucoup d'organisations semblent considérer ces technologies comme un service de messagerie plus rapide, au lieu de voir l'opportunité de disposer d'une infrastructure adaptée à des initiatives stratégiques telles que le « business reengineering » ». Le taux relativement bas d'acceptation des technologies basées sur les télécommunications et en particulier sur l'EDI met en évidence le besoin de former plus profondément les professionnels actuels et futurs dans ce domaine. De telles activités sont encouragées par la publicité croissante qui leur est accordée par l'intermédiaire de la propagande des super-autoroutes des télécommunications reprise par les médias électroniques. » [PARKER, 1994, p 292]

C'est donc dans cette perspective de promotion de l'EDI que les deux universités ont mis sur pied cette collaboration en 1993. Il s'agissait d'estimer à quel point les technologies de communication entre organisations peuvent s'appliquer stratégiquement dans le monde des affaires.

Les organisations mises en jeu étaient des usines imaginaires de grille-pain qui devaient faire appel à leurs fournisseurs et à leurs clients. Les étudiants jouaient le rôle de gestionnaires aux commandes de sociétés impliquées dans des échanges commerciaux via le réseau Internet. L'usage d'Internet a, entre autres particularités, l'avantage d'être gratuit, ce qui réduit drastiquement les coûts liés aux transactions. De plus, le choix de cet environnement est tout à fait réaliste puisque nombre d'universités sont interconnectées via Internet.

Voyons à présent comment ces deux universités ont implémenté leur système EDI. La simulation comportait des logiciels EDI différents: STX pour Curtin, EDI-TIE et ATLAS-EDI pour Maribor (ce dernier est décrit en détail dans ce travail). Ainsi, les étudiants ont pu tester l'interopérabilité de systèmes EDI différents.

2.1.1.1. Système EDI élaboré à Curtin

A Curtin, la sélection d'un système EDI approprié a été guidée par les critères suivants:

- logiciel disponible sur plate-forme PC;
- existence en son sein des fonctionnalités EDI essentielles (incluant la gestion de document, la traduction, la définition des caractéristiques du partenaire commercial,...);
- prise en charge par celui-ci de l'encodage de documents externes, (tels que des bons de commande, des factures...), puisqu'il n'y avait aucune application interne susceptible de fournir les fichiers correspondants;
- présence d'un environnement convivial;
- logiciel supportant le standard EDIFACT, puisqu'il s'agit du standard le plus universel de la communication inter-entreprises.

C'est en fonction de ces exigences que l'université de Curtin a choisi le traducteur STX.

STX a été installé sur le serveur du réseau Novell de l'université, ce qui permet son utilisation à partir de chaque PC du réseau. Le département informatique de l'université de Curtin est relié au réseau Internet via un système UNIX. Il était initialement prévu que l'expédition et la réception des messages liés aux transactions EDI se fasse automatiquement à partir du logiciel STX. Pour cela, il est nécessaire que STX puisse gérer à distance les fonctions de messagerie électronique offertes par le système UNIX. Pour des raisons techniques, les étudiants n'ont pas pu réaliser cette interface entre STX et l'environnement UNIX. Par conséquent, l'expédition et la réception des messages EDI a été réalisée à l'aide d'un logiciel de

messagerie électronique indépendant. Comme le font remarquer les auteurs, cette situation n'est pas représentative de la réalité puisque dans la plupart des installations EDI, les utilisateurs ont recours aux services d'un VAN accessible automatiquement à partir du système EDI.

La figure 2.1 illustre l'architecture des différents systèmes informatiques de l'université de Curtin impliqués dans les échanges EDI.

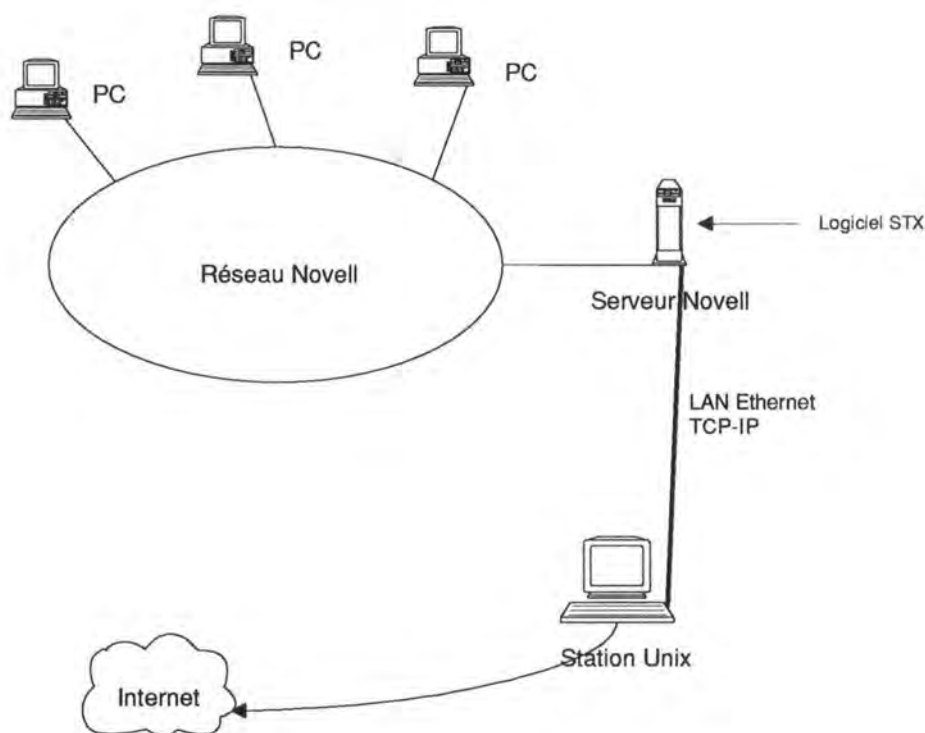


Figure 2.1: Systèmes informatiques impliqués dans les échanges EDI à l'université de Curtin

2.1.1.2. Mise en oeuvre de l'EDI à Maribor

Du côté slovène (Maribor), deux traducteurs EDI ont été utilisés afin de comparer leur utilisation. Il s'agit de:

- ATLAS EDI, développé par la firme Telesmart API Limited en Grande-Bretagne,
- EDI-TIE, développé par EDI-TIE BV aux Pays-Bas.

Ces deux logiciels satisfont les conditions spécifiées par l'équipe de Curtin. Bien qu'ATLAS EDI soit une version mono-utilisateur, il a été installé sur un serveur de fichiers Novell afin d'en assurer l'accès par chaque PC du laboratoire. Quant au logiciel EDI-TIE, il a été installé sur le disque dur d'un des PC du réseau.

L'accès à Internet s'est effectué par l'intermédiaire du système de messagerie SMAIL 400 (conforme à X400) offert par les PTT slovènes. Bien qu'un accès direct

à Internet soit disponible dans cette université, cette solution a été adoptée car elle correspond mieux aux possibilités de communication offertes aux entreprises slovènes.

La figure 2.2 illustre l'architecture des différents systèmes informatiques impliqués dans les échanges EDI de l'université de Maribor.

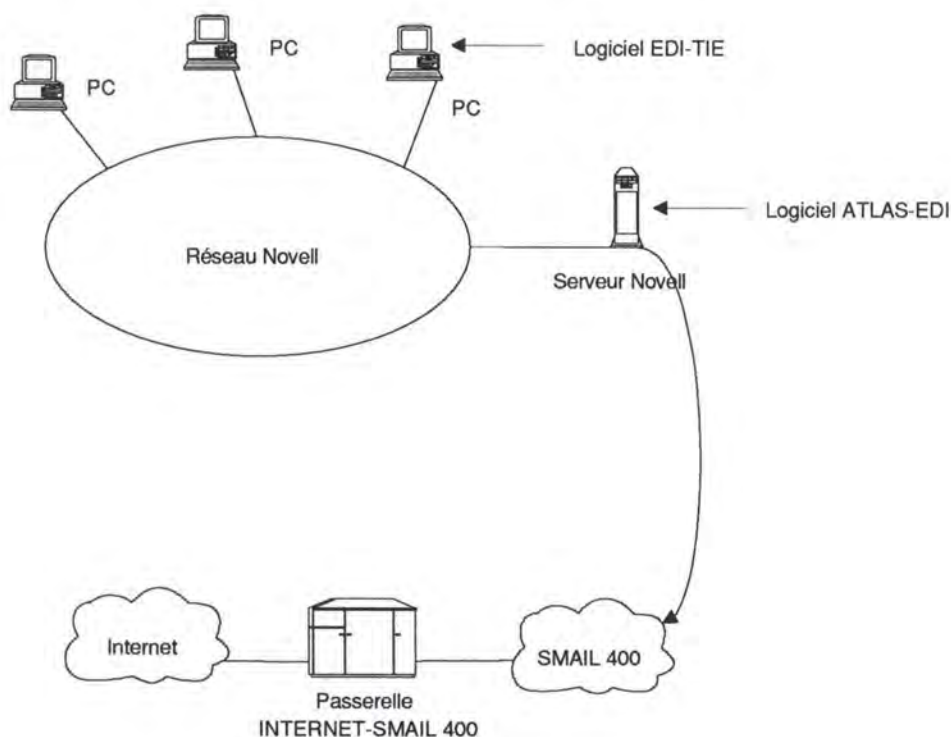


Figure 2.2: Systèmes informatiques impliqués dans les échanges EDI de l'université de Maribor.

Actuellement, ces deux universités préparent pour le début de l'année académique 1995-1996 une nouvelle simulation ayant notamment comme objectif une redéfinition de la méthode à suivre pour concevoir le contenu des messages échangés.

2.1.2. Objectifs par rapports à la simulation

Notre objectif principal étant d'analyser la mise en oeuvre d'un système EDI à l'institut d'Informatique de Namur, l'expérience que nous venons de décrire s'est imposée comme un modèle très intéressant à suivre. Aussi, nous nous sommes d'abord mis en contact avec Craig Parker (Australie) afin de définir les grandes lignes de notre collaboration. Nous avons décidé d'élaborer une méthode à suivre pour échanger des messages EDI semblables à ceux utilisés lors de la simulation entre Curtin et Maribor. La première étape fut donc de choisir un système EDI. Nos critères de choix furent identiques à ceux des équipes de Curtin et Maribor. C'est finalement le logiciel ATLAS EDI qui a remporté notre préférence. Il a été

commandé à la société OFFIS S.A. qui distribue ce produit en Belgique. C'est finalement Monsieur Pascal Waterkeyn, responsable EDI chez OFFIS, qui nous en a fait parvenir la version 2.2.0 de juillet 1994 à la fin du mois de novembre 1994. Notons que dans le courant du mois de février 1995, à la suite d'un entretien informatif, Monsieur Waterkeyn nous a également fourni gracieusement une version démo d'un traducteur EDI, le Mapper d'Interbridge. Ceci nous a permis de réaliser une comparaison des possibilités de traduction offertes par ATLAS et par le Mapper d'Interbridge.

Dès que nous avons installé le logiciel, il nous a paru nécessaire de contacter la deuxième équipe ayant participé à la simulation, celle de Maribor dirigée par Darjan Petric. En effet, comme ils utilisaient le même logiciel que le nôtre, il fut possible de leur demander des explications sur sa configuration.

La configuration du logiciel, l'analyse de son fonctionnement et la proposition d'une démarche pour établir des échanges EDI constituent la substance de notre étude. Nous voulons expliquer les choix qui se posent et décrire les problèmes qui surviennent lors de la mise en oeuvre des échanges. Une discussion sur l'intérêt de recourir à l'EDI dans un tel contexte s'impose également.

La situation que nous décrivons est en fait la mise en application d'une certaine forme d'intégration de niveau un selon [GERSON, 1991] décrite dans le point 1.6.1. En effet, nous utilisons un système de type papier puisque nous ne disposons d'aucune application préalable.

Dans une perspective technique, nos objectifs sont les suivants:

- présenter une démarche pour effectuer une bonne configuration du logiciel en fonction du type des messages échangés (création des tables de conversion entre fichiers plats et EDIFACT);
- présenter une comparaison des possibilités de traduction offertes par ATLAS et par le Mapper d'Interbridge;
- détailler les étapes principales du chemin suivi par les messages lors de leur transmission;
- proposer une interface conviviale de saisie et de présentation de données (développée en Visual Basic) destinée à construire et à lire des fichiers plats correspondant à un type de message EDIFACT.

L'étape de configuration mérite d'être bien expliquée parce qu'elle demande assez bien de temps lorsque l'utilisateur est novice. Nous présenterons une démarche qui peut servir de méthode (il est à noter que la documentation fournie avec le logiciel se borne à décrire des procédures séparément sans jamais suggérer de marche à suivre pour réaliser les manipulations).

Nous proposerons également une interface de saisie de données développée en Visual Basic plus ergonomique que celle proposée par ATLAS.

Afin que la transmission de messages EDI ne constitue pas une boîte noire aux yeux de l'utilisateur, nous décrirons les principales étapes suivies par les données au cours de leur cheminement depuis le PC jusqu'à Internet.

2.2. Structure de l'exposé

L'exposé qui va suivre est structuré selon les objectifs présentés ci-dessus.

Ainsi, nous passerons en revue les points suivants, liés aux objectifs cités:

- analyse des choix à réaliser pour mettre en oeuvre une relation EDI,
- classification des logiciels EDI et présentation d'ATLAS EDI,
- analyse des étapes à suivre pour configurer le logiciel,
- présentation d'exemples de traduction de messages EDIFACT,
- analyse du processus de traduction réalisé par ATLAS EDI et présentation du « Mapper » d'Interbridge,
- synthèse des opérations à réaliser pour échanger des messages avec un partenaire,
- présentation d'une interface fenêtrée qui permet de manipuler des fichiers plats,
- analyse du chemin des données au cours des échanges EDI,
- synthèse générale et discussion.

Mais, avant de développer ces points, il est très important d'effectuer une brève présentation du standard EDIFACT, le seul standard dont il est question dans ce travail. En effet, il est indispensable de bien comprendre la structure des messages EDIFACT (syntaxe et structure des messages) que nous prenons comme exemples. De plus, la configuration du logiciel requiert de l'utilisateur une parfaite maîtrise des éléments syntaxiques d'EDIFACT.

Chapitre 3: La norme EDIFACT et ses composantes

Dans ce qui suit, le lecteur trouvera la définition des termes employés tout au long du travail. Ce chapitre est donc destiné à clarifier certaines notions indispensables notamment à la bonne compréhension de la partie consacrée à la configuration du logiciel ATLAS EDI.

Le lecteur curieux pourra trouver dans les mémoires de [GITS, 1993] et [LEBEAU, 1993] de plus amples informations sur la norme EDIFACT.

L'UN/EDIFACT (United Nations rules for Electronic Data Interchange for Administration, Commerce and Transport) comprend une série de standards, de répertoires et de recommandations pour l'échange électronique de données structurées. Les règles définies par EDIFACT sont publiées dans le UNTDID (United Nations Trade Data Interchange Directory). L'UNTDID comprend notamment:

- les règles de syntaxe EDIFACT au niveau application (Norme ISO 9735);
- le répertoire des éléments de données EDED (EDIFACT Data Element Directory);
- le répertoire des éléments de données composites EDCD (EDIFACT Composite Data Element Directory);
- le répertoire des segments EDSO (EDIFACT Data Segments Directory);
- le répertoire des messages standards des nations unies (UNSM) EDMD (UN/EDIFACT Message Directory).

La syntaxe EDIFACT a été conçue pour qu'un groupe d'utilisateurs (partenaires) puissent échanger des données entre leurs applications. Pour que les échanges soient valides, ils doivent satisfaire à trois conditions au moins:

- d'une part, les données échangées doivent être conformes à leur définition établie dans le EDED;
- d'autre part, leur structure doit obéir aux règles syntaxiques spécifiées par la norme ISO 9735. En effet, c'est la position de chaque élément de donnée dans un message qui précise la nature et le contenu sémantique de la valeur transmise;
- enfin, les partenaires doivent s'être préalablement mis d'accord sur une représentation commune des données qu'ils s'échangent (type de documents échangés, version d'EDIFACT utilisée,...).

Ce dernier point constitue la pierre angulaire de cette étude et sera donc développé plus tard.

Tout au long de l'exposé, nous allons utiliser les termes d'interchanges et de messages. Aussi, il est bon de clarifier ces deux notions une fois pour toutes en expliquant brièvement les relations qui existent entre elles.

-et les caractères « + », « : », « ? » usuellement utilisés comme caractères de service (séparateurs ou indicateurs) et compatibles avec le TELEX international,

D'autres caractères sont également admis par la syntaxe de niveau A et sont incompatibles avec le TELEX: « ! », « & », « % », « * », « ' ' », « < », « > ».

3.2. Les éléments de données

Il existe deux types d'éléments de données: les éléments de données simples et composites. Les premiers seront appelés simplement éléments de données.

3.2.1. Les éléments de données (simples)

Au sein du EDED, chaque élément de donnée est défini par (cf. figure 3.1):

- un indicatif numérique (noté CODE ou TAG) et une désignation alphabétique,
- une description sémantique,
- le type de représentation de l'élément de donnée,
- une note (observation supplémentaire),

Indicatif numérique et désignation	3413 Departement or employee identification
Description sémantique	Desc: Internal identification code.
Type de représentation	Repr: an..17
Note	Note: Code specified by organisation concerned

Figure 3.1: Exemple de définition d'un élément de donnée au sein du EDED.

L'indicatif numérique d'un élément de donnée est composé de quatre chiffres qui forment un code de classification des données.

La représentation d'un élément de donnée peut être:

- soit alphabétique (a),
- soit numérique (n),
- soit alphanumérique (an).

Le type de représentation définit également la taille maximum de chaque élément de donnée:

- a2 signifie que l'élément de donnée est alphabétique et de longueur 2,
- an..17 signifie que l'élément de donnée est alphanumérique et de longueur maximale 17.

3.2.2. Les éléments de données composites

Un élément de donnée composite est un regroupement ordonné de données élémentaires formant un ensemble cohérent. Dans le EDCD (cf. figure 3.2), l'indicatif d'une donnée composite commence par la lettre C suivie de 3 chiffres. Les données constitutives d'une donnée composite sont, soit obligatoires (M pour « mandatory »), soit facultatives (C pour « conditional »).

Indicatif numérique et désignation	C056 Department or employee details
Description sémantique	Desc: Code and/or name of a department or employee. Code preferred
Éléments de données constitutifs	3413 Department or employee identification c an..17 3412 Department or employee c an..35

Figure 3.2: Exemple de définition d'un élément de donnée composite au sein du EDCD.

Le principe du codage des éléments de données composites est exposé dans le paragraphe 3.2.5.

3.2.3. Données codées

Un grand nombre d'éléments de données simples sont représentés par des codes dont la signification est définie dans le TDED.

3.2.4. Qualifiants et données qualifiées

Un qualifiant est un élément de donnée précisant la sémantique d'un autre élément de donnée générique. L'utilisation de qualifiants est d'un grand intérêt. Elle permet de diminuer le nombre de données à manipuler et d'enrichir la valeur sémantique d'une donnée qualifiée par l'adjonction de nouvelles valeurs qualifiantes. Les listes de valeurs associées à une donnée qualifiante sont enregistrées dans le répertoire des codes (Code Sets Directory). Notons que les qualifiants sont également utilisés pour préciser la sémantique de certains segments.

3.3. La notion de segments de données EDIFACT

Un segment est une suite ordonnée d'éléments de données simples ou composites obligatoires ou facultatifs. Il faut considérer les segments comme des groupes de données se rapportant à une même entité. Par exemple, les éléments de données du segment NAD (Name And Address) définissent l'identité et l'adresse du partenaire spécifié dans ce segment. Pour rappel, le répertoire des segments standards recommandés par les Nations unies porte le nom de EDSD (EDIFACT Segment Directory).

3.3.1 Structure d'un segment

Nous faisons la distinction entre la notion de segment et la notion de codage d'un segment.

Un segment est défini par:

- une étiquette (TAG);
- un intitulé de segment (n'apparaît pas dans le codage);
- une fonction (n'apparaît pas dans le codage);
- une suite d'éléments de données simples ou composites, chacun suivi de son statut (obligatoire (M) ou facultatif (C)) et de son type de représentation (dans le cas d'un élément de donnée simple).

TAG	FTX
Intitulé de segment	Free Text
Fonction	To provide free form or coded text information
Eléments de données constitutifs (simples et composites)	4451 Text subject qualifier M an..3 4453 Text Function, coded C an..3 C107 Text Reference C C108 Text literal C

Figure 3.3: Exemple de définition d'un segment de donnée.

Au niveau de son codage, un segment est représenté par:

- une en-tête: il s'agit d'un élément de donnée composite. Le premier élément de donnée de l'en-tête est l'étiquette de segment (TAG) représentée par trois caractères alphabétiques. Dans la plupart des cas, cet élément de donnée (obligatoire) est le seul mentionné. Les autres éléments de données (facultatifs) qui suivent l'étiquette du segment dans l'en-tête sont présentés plus loin;
- une suite d'éléments de données simples ou composites. Au niveau du codage ils seront délimités par différents caractères séparateurs.

Nous reviendrons plus en détail sur le codage des segments dans le paragraphe 3.5.2.

Il est important de rappeler que, lorsqu'une donnée composite est obligatoire, au moins une des données constitutives doit apparaître au niveau du codage du segment. Lorsque la donnée composite est facultative, toutes les données constitutives peuvent être absentes, y compris celles qui sont indiquées comme obligatoires dans la définition de la donnée composite. Lorsque la donnée composite

est présente, c'est-à-dire lorsqu'au moins une des données constitutives apparaît, alors toutes les données constitutives obligatoires doivent être présentes.

Deux catégories de segments sont à distinguer: les segments de données de l'application de l'utilisateur (Data Segments) qui contiennent les informations internes au message (ex: noms et adresses, dates, montants, quantités, ...) et les segments de contrôle (Control Segments) qui définissent le cadre et les caractéristiques de l'interchange EDIFACT.

3.3.2. Les segments de contrôle et la structure de l'interchange

Les segments de contrôle contiennent des éléments de données de service tels que l'émetteur de la transmission, le type et le niveau de la syntaxe, la date de préparation, le degré de priorité etc... L'étiquette (TAG) des segments de contrôle commence toujours par les deux caractères « UN ». Ces deux caractères de début de TAG de segment sont exclusivement réservés aux segments de contrôle. La figure 3.4 illustre la structure générale d'un interchange, avec ses segments de contrôle (dont l'étiquette commence par « UN ») et ses segments de données (dont l'étiquette est représentée ici par les lettres XXX qui peuvent prendre les valeurs fixées par la norme, par exemple « FTX »):

INTITULE DU SEGMENT	TAG	STATUT (C/M)
Avis de chaîne de caractères de service	UNA	C
Segment d'en-tête du contrôle d'échange	UNB	M
En-tête de groupes fonctionnels	UNG	C
En-tête de message	UNH	M
Segments de données utilisateurs internes au message	XXX	
Segment de délimitation D du message	UNS	C
Segment de données utilisateurs internes au message	XXX	
Segment de délimitation S du message	UNS	C
Segments de données utilisateurs internes au message	XXX	
Fin de message	UNT	M
Fin de groupe fonctionnel	UNE	C
Fin d'interchange	UNZ	M

Figure 3.4: Structure générale d'un interchange, avec ses segments de contrôle.

La structure de l'interchange présenté ci-dessus comporte un seul groupe fonctionnel (délimité par les segments UNG et UNE) contenant un seul message

(délimité par les segments UNH et UNT). Un interchange peut contenir plusieurs groupes fonctionnels et un groupe fonctionnel plusieurs messages codés sur la même structure générique de message EDIFACT. Une transmission peut, elle, contenir plusieurs interchanges. Le rôle des différents segments de contrôle est le suivant:

3.3.2.1. Le segment UNA

C'est le segment qui permet de préciser les caractères choisis comme séparateurs et indicateurs dans l'interchange. Ce segment n'est pas obligatoire. Les caractères séparateurs usuels (utilisés par défaut si le segment UNA est absent) diffèrent selon le niveau A ou B de syntaxe utilisé. L'utilisateur peut choisir d'autres caractères de service à sa convenance. La forme du segment UNA usuel en syntaxe de niveau A est:

UNA:+.?'

Ce segment indique que les conventions suivantes sont adoptées pour l'interchange considéré:

- le séparateur d'éléments constitutifs d'un élément composite est « : »;
- le séparateur entre éléments de données du segment est « + »;
- le caractère suspensif est « ? » (précédant un caractère séparateur, il indique que celui-ci ne doit pas être considéré comme caractère séparateur);
- le caractère de fin de segment est « ' ».

3.2.2.2. Les segments UNB et UNZ

Le segment UNB, en-tête de contrôle de l'interchange, est un segment obligatoire qui a pour but d'initialiser, identifier et préciser le cadre d'un interchange. Les données du segment comportent obligatoirement les données composites suivantes:

- un identifiant et un numéro de version de la syntaxe (par exemple, la donnée UNOA signifie « syntaxe de niveau A »)
- l'identification de l'émetteur de l'échange,
- l'identifiant du destinataire,
- la date et l'heure de la transmission,
- la référence de contrôle de l'interchange (élément de donnée simple) et les éléments de données facultatifs tels que: mot de passe, référence de l'application, priorité de traitement, demande d'accusé de réception, identification de l'accord de communication, indicateur d'essai.

Le segment UNZ de fin d'interchange, obligatoire, est destiné à terminer et vérifier l'achèvement d'un interchange. Il contient deux éléments obligatoires, un compteur du nombre de messages (ou de groupes fonctionnels, si utilisés) de l'interchange et un rappel de la référence de contrôle de l'interchange.

3.3.2.3. Les segments UNG et UNE:

Le segment UNG initialise et identifie un groupe fonctionnel, c'est-à-dire un ensemble de messages du même type regroupés dans l'interchange. Ce segment n'est pas obligatoire et comporte comme données:

- une identification du groupe fonctionnel, qui identifie le type de message,
- un identifiant de l'émetteur,
- un identifiant du récepteur,
- la date et l'heure de préparation,
- le numéro de référence du groupe fonctionnel attribué par l'émetteur,
- le code de l'agence de contrôle de l'instance normative,
- le numéro de version du message,
- un mot de passe du destinataire (facultatif).

Le segment UNE est destiné à terminer et à vérifier l'achèvement d'un groupe fonctionnel. Il comporte deux éléments: un compteur du nombre de messages du groupe fonctionnel et la référence du groupe fonctionnel attribuée par l'émetteur, la même que celle figurant dans l'UNG correspondant.

Entre les segments UNG et UNE correspondants sont insérés les segments des messages qui se succèdent dans l'ordre spécifié par la structure du message utilisé. Plusieurs groupes fonctionnels peuvent se succéder à l'intérieur de l'interchange, et plusieurs interchanges dans une même transmission.

3.3.2.4. Les segments UNH, UNT et UNS

Comme ces segments de contrôle sont spécifiques aux messages, ils sont définis au point 3.4.2. Notons déjà que les segments UNH et UNT définissent respectivement le début et la fin d'un message.

Le schéma hiérarchique d'une transmission en interchange EDIFACT est représenté à la figure 3.5.

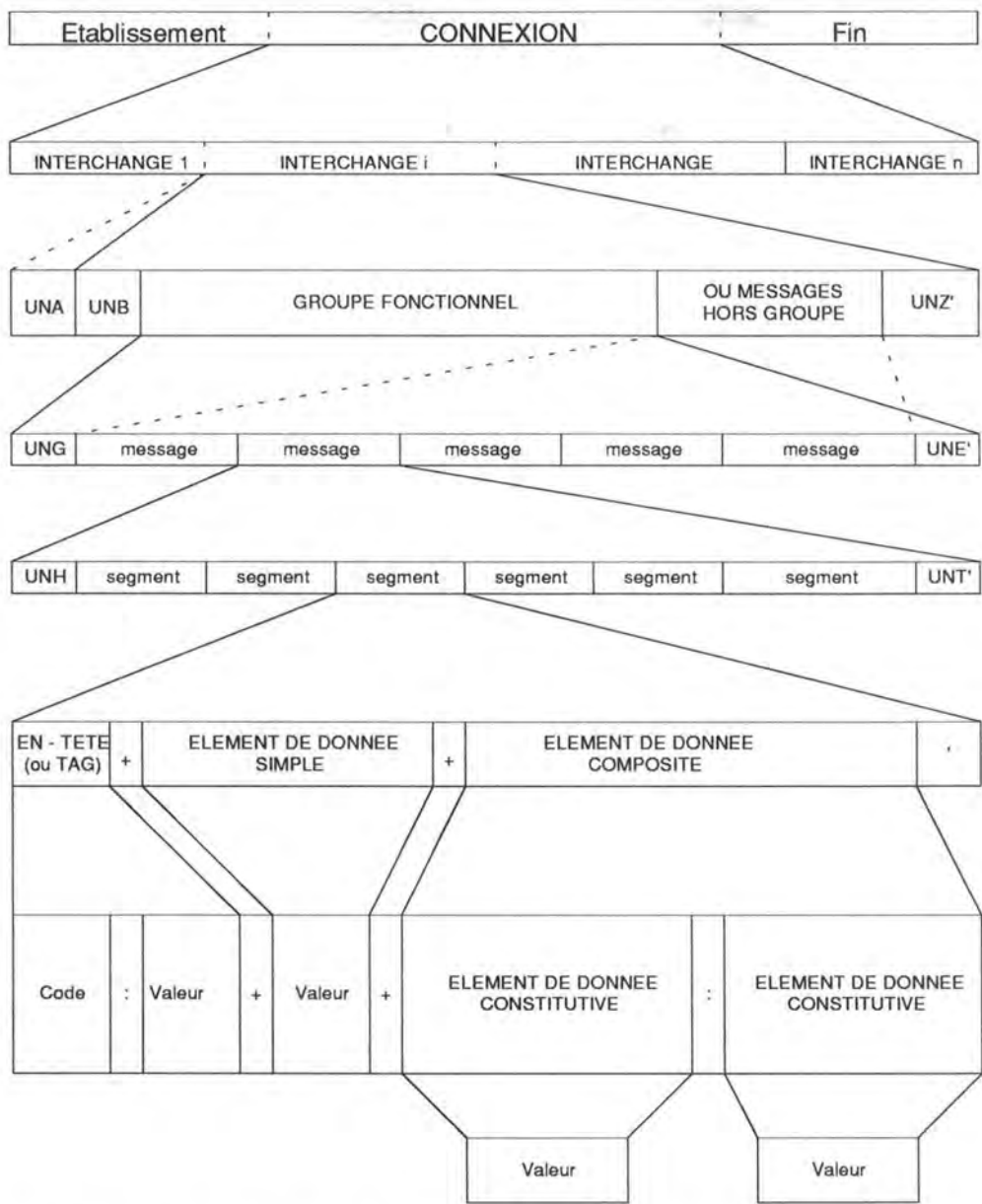


Figure 3.5: Schéma hiérarchique d’une transmission en interchange EDIFACT

3.4. La structure des messages EDIFACT

Plus haut, nous avons décrit la structure d’un interchange EDIFACT. A présent, décrivons la structure d’un message EDIFACT.

Un message codé en EDIFACT est composé d’une suite de segments ordonnés selon une structure arborescente à plusieurs niveaux d’imbrication dont les sommets sont étiquetés par les étiquettes (TAGS) des segments du message. Les niveaux des segments dans la structure arborescente augmentent avec la profondeur du segment dans l’arborescence.

L'ordre d'apparition des segments dans le message codé est nécessairement conforme à celui indiqué dans le parcours en profondeur d'abord et de gauche à droite des segments dans le diagramme du message, description arborescente du message.

3.4.1. Description du diagramme d'un message

Dans le diagramme, les segments sont identifiés par leur étiquette (TAG); le diagramme comporte obligatoirement le segment d'en-tête UNH et le segment de fin UNT.

Le statut de chaque segment dans le message, c'est-à-dire la nécessité de son apparition au moins une fois ou non dans le message codé, est indiqué au-dessous de l'étiquette par la lettre M (Mandatory) ou C (Conditional) suivi immédiatement par le nombre de répétitions. Un segment obligatoire devra apparaître dans le message codé au moins une fois et au maximum un nombre de fois limité par une valeur spécifiée. Un segment facultatif peut être omis ou apparaître un certain nombre de fois (limité aussi par une valeur maximale).

3.4.1.1. Emboîtement vertical des segments

L'occurrence d'un segment peut être liée à celle d'un autre segment situé à un niveau inférieur dans la structure arborescente décrivant le message. On dit dans ce cas que le segment est imbriqué (ou emboîté) dans l'autre.

Les segments de niveau 0 ne peuvent ni se répéter, ni contenir des segments imbriqués.

3.4.1.2. Groupage des segments

Deux segments ou plus peuvent être groupés. Le segment dans lequel les autres s'imbriquent (de numéro de niveau le plus bas) est appelé segment de contrôle du groupe. Ce segment est unique. Deux groupes distincts ne peuvent débiter par un même segment de contrôle. Les groupes possèdent également un statut (M ou C) et un nombre de répétitions. Les groupes peuvent également être imbriqués, un groupe peut contenir un ou plusieurs groupes de niveau supérieur.

Examinons l'exemple de la figure 3.6 illustrant un message fictif.

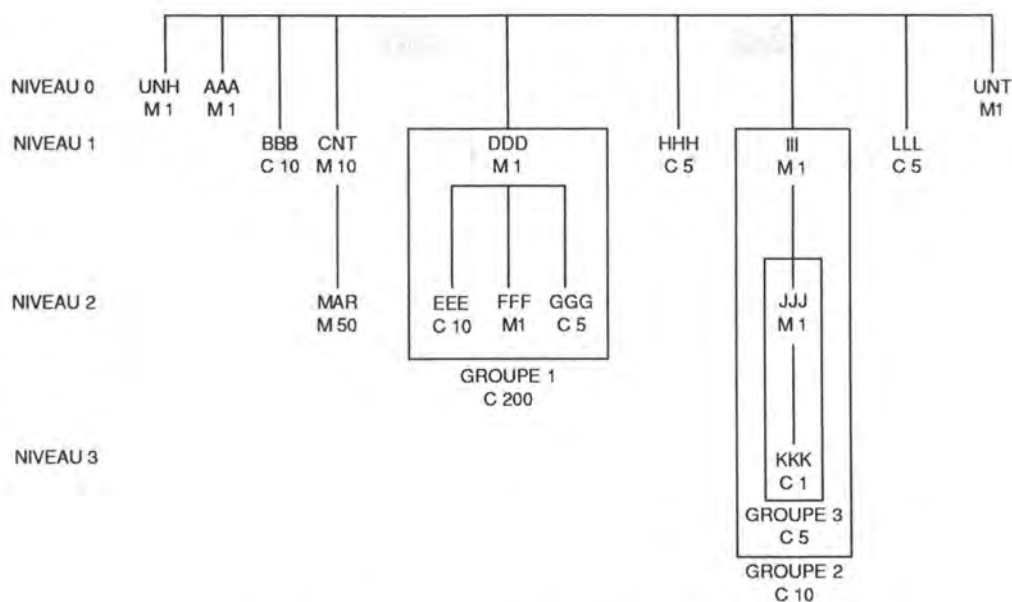


Figure 3.6: Diagramme d'un message fictif.

Remarquons dans cette structure l'emboîtement vertical du segment MAR répétitif obligatoire dans le segment répétitif obligatoire CNT. Pour montrer l'intérêt de cet emboîtement de manière concrète, imaginons que le segment CNT contienne les données décrivant un conteneur d'une expédition de marchandises et que les segments MAR contiennent les données correspondant aux cinquante types d'articles pouvant être stockés dans le conteneur. Le diagramme indique que pour chaque apparition d'un segment CNT (qui peut apparaître dix fois au maximum) le segment MAR peut se répéter entre une et cinquante fois. Le segment MAR répétable est bien emboîté dans le segment répétable CNT.

Nous remarquons également que les segments « frères » EEE, FFF et GGG sont imbriqués dans le segment DDD, leur « père » commun. Ceci implique qu'après l'apparition dans le message codé d'un segment DDD (segment de contrôle obligatoire du groupe 1) peuvent apparaître entre zéro et dix segments EEE, suivis nécessairement d'un segment FFF, suivi lui-même d'au maximum cinq segments GGG. De plus, ce groupe 1 est conditionnel et peut apparaître entre zéro et deux cents fois consécutivement dans le message codé.

Le groupe 2 dont le segment de contrôle est III contient le groupe 3 de segment de contrôle JJJ. Le groupe 3 est imbriqué dans le groupe 2. Dans le codage du message, après chaque apparition du segment III, peuvent apparaître les segments du groupe 3 (de zéro à cinq fois). En supposant, par exemple, que le message codé laisse apparaître deux fois les groupes 1 et 3 et que les autres segments et groupes apparaissent exactement une fois, la séquence des segments (en respectant le parcours en profondeur d'abord et de gauche à droite) est la suivante :

UNH, AAA, BBB, CNT, MAR, DDD, EEE, FFF, GGG, DDD, EEE, FFF, GGG, HHH, III, JJJ, KKK, JJJ, KKK, LLL, UNT.

3.4.2. Les segments de contrôle d'un message

Les segments de contrôle intervenant dans un message sont, d'une part, les segments UNH et UNT respectivement segments de contrôle en-tête et fin de message qui apparaissent nécessairement dans le diagramme de tout message et, d'autre part, les segments de contrôle de section UNS qui peuvent apparaître au plus deux fois dans un diagramme de message.

3.4.2.1. Le segment UNH

Il contient deux éléments de données obligatoires:

- le numéro de référence du message,
- l'identificateur du message,

et deux éléments de données conditionnels:

- la référence commune d'accès,
- le statut du transfert.

Le numéro de référence est généralement un numéro généré par programme qui correspond à la position du message dans la séquence des messages dans l'interchange ou dans un groupe fonctionnel de l'interchange.

L'identificateur de message est une donnée composite comprenant 5 éléments de données constitutives qui sont:

- le type de message (obligatoire - donnée de 6 caractères maximum),
- le numéro de version du message (obligatoire - donnée de 3 caractères maximum),
- le numéro d'édition du message (ou numéro de release, conditionnel),
- l'organisme de contrôle dont dépend le message (obligatoire dans l'UNH uniquement lorsque le groupage fonctionnel n'est pas utilisé);
- le code attribué par l'association (conditionnel).

3.4.2.2. Le segment UNT

Le segment de contrôle de fin de message UNT contient deux éléments de données de service obligatoires.

La première donnée indique le nombre de segments du message (y compris les segments UNH et UNT).

La seconde, numéro de référence du message, contient la référence du message identique à celle figurant dans le segment UNH.

3.4.2.3. Les segments UNS

Les segments de contrôle de section sont utilisés pour délimiter des sections dans un message et contiennent un seul élément de donnée d'identification de section. Deux segments UNS au maximum peuvent être introduits dans le diagramme de description d'un message. Ces segments sont non répétables et leur apparition est facultative dans le message codé. Dans le cas où deux segments UNS sont utilisés, le premier délimite une section en-tête de message et une section détail du message et contient la valeur « D » (Details Section), le second délimite la « section détails » par rapport à une section récapitulative et contient la valeur « S » (Summary Section).

3.5. La syntaxe de transfert des fichiers d'interchange EDIFACT

Le codage d'un interchange est constitué d'une suite ordonnée de segments codés, en conformité avec la structure générique de l'interchange décrite ci-dessus et avec la structure des messages contenus dans l'interchange EDIFACT.

3.5.1. Codage d'un message

Le codage d'un message contenu dans l'interchange est constitué d'une suite ordonnée de segments codés, en conformité avec l'ordre de lecture du diagramme descriptif du message (parcours en profondeur d'abord, c'est-à-dire lecture des segments en priorité de haut en bas et de gauche à droite).

Deux types de codage existent pour les messages, le codage implicite et le codage explicite.

Dans le premier cas, tous les segments du message qui sont susceptibles de se répéter à l'intérieur du message sont représentés de manière implicite, c'est-à-dire sans adjonction des numéros de contrôle associés à l'étiquette du segment, informations qui précisent les numéros de répétition du segment et de ses ascendants à chaque niveau d'imbrication de celui-ci. Dans le second cas, au contraire, tous les segments du message qui sont susceptibles de se répéter à l'intérieur du message sont représentés de manière explicite avec, associé à l'étiquette du segment, le codage des numéros de contrôle qui précisent la répétition du segment et de ses ascendants à chaque niveau d'imbrication de celui-ci.

Un message normalisé par les instances de l'EDIFACT est spécifié exclusivement soit pour le codage implicite, soit pour le codage explicite. La plupart des messages normalisés actuellement utilisent le codage implicite.

3.5.2. Codage d'un segment et règles de compression dans le codage et la transmission d'un segment

Le codage d'un segment consiste en la suite ordonnée des codages des données simples ou composites qui le composent, chaque donnée étant séparée par le caractère séparateur de données, généralement le caractère « + ». L'étiquette de l'en-tête du segment est obligatoire et la dernière donnée du segment est suivie du

caractère spécifique de fin de segment, généralement le caractère « ' ». L'ordre des données dans le segment est décrit dans la description du segment dans le répertoire de segment EDSO. Le codage générique est donc de la forme:

TAG+DE1+DE2+...+DEn'

où DEi est un élément de donnée d'ordre i (pour i variant de 1 à n).

La figure 3.7 nous en donne un exemple trivial.



Figure 3.7: Exemple de codage d'un segment de donnée.

Le codage d'une donnée simple est fonction de son type de représentation décrit dans le EDSO. Le codage d'une donnée composite est la suite des codages des données qui la constituent, chaque donnée constitutive étant séparée par le caractère séparateur de données constitutives, généralement le caractère « : ».

Le codage générique d'un segment comportant une donnée composite constituée de 3 données constitutives (en prenant le sigle CE pour Constitutive Element) est de la forme:

TAG+DE1+DE2+CE31:CE32:CE33+DE4+.....DEn'

La figure 3.8 nous en donne un exemple.

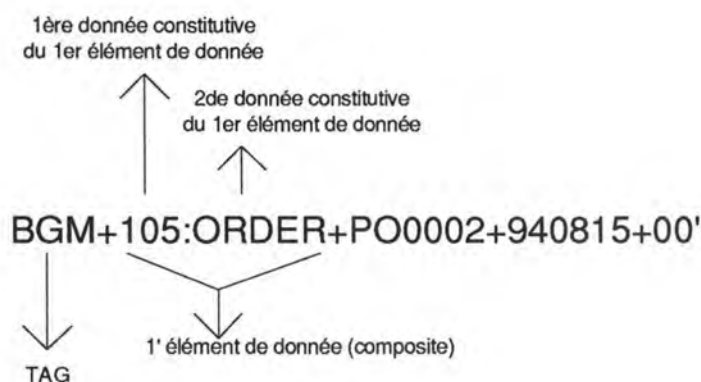


Figure 3.8: Exemple de codage d'un élément de donnée composite.

Les techniques de compression sont définies par une série de règles:

Règle 1: Omission des éléments de données.

Si un élément de donnée de statut C dans la description du segment est omis et est suivi d'un autre élément de donnée, le caractère séparateur de données est transmis en lieu et place de l'élément de donnée considéré. Dans l'exemple suivant deux éléments de données (DE2 et DE3) ont été omis:

TAG+DE1+++DE4+DE5+DE6+DE7+DE8'

Règle 2: Troncation des éléments de données.

Si des éléments de données sont omis en fin de segment, celui-ci peut être tronqué par le caractère de fin de segment. Ainsi, dans l'exemple suivant, les trois derniers éléments de données de l'exemple précédent ont été omis en fin de segment:

TAG+DE1+++DE4+DE5'

Règle 3: Exclusion de segments.

Les segments de statut C dans le diagramme du message qui ne comportent aucune des données doivent être omis totalement y compris leur étiquette d'en-tête.

Règle 4: Omission d'éléments de données constitutifs.

Lorsqu'un élément de donnée constitutif est omis dans un élément composite, le caractère séparateur d'éléments de données constitutives est transmis en place et en lieu de l'élément omis.

Dans l'exemple suivant, trois éléments de données constitutives ont été omis, le premier (CE21) au début du second élément de donnée du segment et les deux autres (CE42 et CE43) dans le quatrième et dernier élément de donnée du segment:

TAG+DE1+:CE22:CE23+DE3+CE41:::CE44'

Règle 5: Troncation des éléments de données constitutifs.

Si les éléments de données constitutifs sont omis en fin de l'élément composite, celui-ci peut être tronqué par le caractère séparateur de données « + » ou par le caractère de fin de segment « ' » selon la position de la donnée composite concernée. Ainsi, dans l'exemple suivant, qui reprend l'exemple précédent en supprimant la dernière donnée constitutive de la donnée numéro 2 (CE23) et la dernière donnée constitutive de la donnée en dernière position dans le segment (CE44), les deux éléments de données composites ont été tronqués (par suppression des caractères séparateurs de données constitutives « : »).

Soit:

TAG+DE1+:CE22+DE3+CE41'

Il faut remarquer que l'élément de donnée en-tête du segment est lui-même un élément composite dont le TAG du segment est la première donnée constitutive (voir ci-après la représentation explicite des segments où les autres données constitutives peuvent apparaître), ainsi dans les exemples que nous venons de présenter, il y a troncation des données constitutives pour l'élément d'en-tête des segments dont le TAG est présent.

Règle 6: Suppression des caractères non significatifs.

Il est recommandé, pour accroître l'efficacité de la transmission et limiter les temps de traitement, de supprimer les caractères non significatifs dans les éléments de données qui ont une représentation de longueur variable. Par exemple, si une donnée numérique est déclarée n..10 et que la valeur associée est 00000012, la donnée figurant dans le message devient ...+12+...

Au terme de cette mise au point des principales caractéristiques d'EDIFACT, nous entrons dans le vif du sujet en abordant les choix qui se posent lorsque l'on désire effectuer des échanges EDI.

Chapitre 4. Analyse des choix à effectuer pour mettre en oeuvre des échanges EDI

La mise en oeuvre d'une relation EDI implique de nombreux choix, non seulement techniques mais aussi concernant la nature des messages EDI à échanger.

Tout au long de ce travail, nous allons mettre en évidence les choix qui se sont imposés afin de réaliser des échanges de messages EDIFACT. La démarche que nous avons suivie trouve son origine et son fondement dans l'exemple que représente la simulation réalisée entre les universités de Curtin et Maribor. Plutôt que de nous borner à reproduire fidèlement la méthode suivie par les deux chercheurs (Craig Parker et Darjan Petric) et leurs étudiants, nous apporterons des commentaires et des remarques en guise de suggestions méthodologiques.

Pour commencer, il est instructif d'analyser succinctement les choix qui se posent à une organisation de type commercial et de les comparer à ceux que nous avons rencontrés. D'entrée de jeu, on verra que le cadre universitaire se différencie nettement du contexte dans lequel sont plongées les entreprises désireuses d'entreprendre des relations EDI avec des partenaires commerciaux.

Afin de structurer la comparaison, nous nous référons au travail réalisé par N. Gits [GITS, 1992]. Ce travail contient notamment le relevé des questions qu'une organisation de type commercial doit se poser préalablement à l'instauration de relations EDI:

- quels partenaires choisir?
- quels sont les bénéfices que l'entreprise peut retirer de l'EDI?
- quel niveau d'intégration choisir pour implémenter l'EDI?
- quels sont les investissements à réaliser pour pratiquer des échanges EDI?
- comment choisir un standard EDI parmi tous ceux qui existent?
- quels documents seront échangés via l'EDI?
- comment choisir un logiciel de traduction EDI?
- quels sont les supports matériels (hardware) dont il faut disposer?
- quels moyens de télécommunication choisir?

Chacune de ces questions est passée en revue et commentée brièvement.

4.1. Choix des partenaires

Une organisation qui désire se lancer dans des échanges EDI doit non seulement étudier les moyens de leur mise en oeuvre mais aussi établir la liste de ses partenaires potentiels. C'est aussi en fonction des caractéristiques des partenaires que le choix des moyens techniques doit s'effectuer.

Pour nous, ce choix a été trivial puisque c'est nous qui avons fait la démarche de contacter Craig Parker et Darjan Petric afin de collaborer avec eux.

4.2. Evaluation des bénéfices apportés par l'EDI

Dans le cadre réel du commerce électronique entre des sociétés consommatrices et productrices de biens à valeur ajoutée, les enjeux économiques sont bien entendu primordiaux et conditionnent le recours à l'EDI et les modalités de sa mise en oeuvre. Ainsi, les sociétés doivent préalablement se poser la question de savoir quels sont les bénéfices liés à l'usage de l'EDI. Il faut également se rendre compte de la largeur du spectre des conséquences de l'EDI pour une entreprise. En effet, l'adoption d'une telle technologie a des répercussions sur bon nombre d'aspects tels que:

- la nature et l'efficacité des traitement des données de toute l'organisation,
- la nature des tâches allouées au personnel,
- la gestion et les coûts de production,
- l'étendue (élargissement) des zones de marché.

Au contraire, dans le contexte d'une simulation universitaire, les enjeux économiques sont inexistantes et s'effacent devant l'apport éducatif que représente la mise en oeuvre de l'EDI et son évaluation.

4.3. Choix d'un niveau d'intégration de l'EDI

Comme nous l'avons vu dans le chapitre 1, il existe différents scénarios d'intégration de l'EDI dans les activités d'une organisation. Bien entendu, le choix et l'adoption d'un de ces scénarios influencent largement les autres facteurs à considérer.

4.4. Estimation des investissements nécessaires à la mise en oeuvre de l'EDI

Les investissements à réaliser par une entreprise pour établir une relation EDI le sont dans les domaines suivants:

- le support matériel nécessaire à la mise en oeuvre du système EDI et à son évolution,
- l'achat ou la création d'un logiciel de traduction EDI,
- l'installation de liens de communication,
- l'adaptation éventuelle des applications et des systèmes de bases de données utilisées par les applications,
- les frais de formation des utilisateurs,
- les frais de personnel affecté à la maintenance du système EDI,
- le coût (en temps) de la négociation avec les partenaires commerciaux.

Il est bien évident que les coûts de matériel et d'établissement de liaisons de communication peuvent être réduits si l'entreprise utilise les installations existantes.

En ce qui nous concerne, nous avons utilisé le système en place à l'Institut d'Informatique (voir à ce propos la partie consacrée au chemin des données). Le seul investissement a été l'achat du logiciel de traduction.

4.5. Choix des standards EDI

Nous avons déjà mentionné l'existence de différents types de standards mis à la disposition des utilisateurs. Il est évident que des partenaires commerciaux ont tout intérêt à adopter le même standard pour la représentation des données qu'ils s'échangent. Les coûts liés à la coordination et à l'intégration de plusieurs standards au sein d'une relation d'échanges peuvent être supprimés par l'adoption d'un même et unique standard dont la diffusion est la plus générale possible. De ce point de vue, EDIFACT est un excellent candidat.

Dans notre travail, nous avons choisi EDIFACT comme standard puisqu'il a été utilisé par les deux universités australienne et slovène. Plus précisément, nous avons adopté une même version d'EDIFACT, la version 90.1, toujours pour la même raison.

4.6. Choix des documents à échanger par l'EDI

Il faut savoir que la plupart du temps, les messages EDI sont calqués sur le modèle des documents papiers traditionnels échangés entre les entreprises. Par conséquent, la nature des documents papiers traditionnels nécessaires à l'établissement d'un échange électronique constitue le premier facteur qui guide la conception des messages EDI.

Comme nous l'avons vu, la simulation menée par les universités de Craig Parker et Darjan Petric était une simulation de relations commerciales. C'est la raison pour laquelle les documents qu'ils ont choisis étaient des bons de commande et des factures. En réalité, ils n'ont pas utilisé les messages standardisés correspondants à ces documents (les UNSM définis par les Nations unies), mais ont conçu des messages personnalisés dont la structure et le contenu en segments ne répondaient qu'à leurs besoins. Nous avons procédé de la même manière, en concevant des messages dont nous présenterons quelques exemples dans le chapitre consacré à la configuration du logiciel de traduction. La composition en segments (ainsi que leur statut obligatoire ou facultatif et les caractéristiques de leurs éventuelles répétitions) des messages échangés doit aussi faire l'objet d'une concertation avec Craig Parker. Nous reviendrons sur ce point plus tard (chapitre 12).

4.7. Choix d'un logiciel de traduction

Le choix d'un logiciel de traduction est une étape cruciale car il doit tenir compte d'une série de paramètres:

- la flexibilité: une entreprise pouvant avoir recours à des standards et des moyens de télécommunication différents (notamment en fonction des partenaires), le logiciel de traduction utilisé doit être capable de les supporter tous;
- l'extensibilité: le logiciel doit fournir la possibilité d'incorporer de nouveaux partenaires rapidement et facilement;
- la compatibilité avec les applications existantes.

Dans notre cas, ces paramètres ne possèdent pas l'importance qu'ils ont dans un milieu industriel et commercial. En effet, nous avons acheté un logiciel sans posséder d'application préalable. En réalité, le logiciel ne devait répondre à aucun besoin commercial. Par conséquent, l'extensibilité du système est un critère qui n'intervient pas dans notre choix. Le critère de flexibilité n'est pas non plus important pour nous puisque nous nous limitons à un seul partenaire qui n'utilise qu'un seul standard et avec lequel nous ne communiquons qu'au moyen du courrier électronique.

Bien entendu, le choix d'un logiciel de traduction dépend également des fonctionnalités offertes par ce logiciel. Nous reviendrons sur ce point plus tard (chapitre 6).

4.8. Choix du matériel

En toute généralité, une organisation doit choisir le type de support matériel qui lui convient le mieux pour accomplir les opérations suivantes:

- la création des fichiers plats destinés à être traduits en fichiers de format EDI,
- la traduction des messages (fonction de convertisseur),
- la transmission des messages EDI.

Une organisation peut utiliser des systèmes mainframe, ou des systèmes basés sur des micro-ordinateurs pour chacune des opérations citées. Comme le choix d'un système pour une opération peut être indépendant de celui pour les deux autres, il existe au plus huit combinaisons possibles.

Dans notre cas, toutes les opérations de création de fichiers plats, de traduction et de communication sont réalisées à partir d'un PC en réseau (voir la partie consacrée au chemin des données).

4.9. Choix d'un moyen de télécommunication

Le choix d'un support de télécommunication par une organisation doit s'effectuer notamment selon les critères suivants:

- la nécessité de transférer les messages sans erreur,
- la nécessité d'assurer la sécurité des transferts d'information,
- la nécessité ou non de disposer d'un système avec accusés de réception.

Dans le cadre de ce travail, nous avons suivi l'exemple de la simulation de Craig Parker et Darjan Petric, en utilisant le courrier électronique transitant par le réseau Internet. Les raisons principales de ce choix sont sa disponibilité immédiate à l'institut d'Informatique et la simplicité de sa mise en oeuvre. On peut y ajouter le fait que le système de notre correspondant ne doit pas être opérationnel en même temps que le nôtre.

Nous venons d'aborder les critères généraux qui doivent guider toute organisation désireuse d'entreprendre une relation EDI. Dans la suite, nous centrons notre intérêt au niveau des logiciels de traduction (et spécialement le nôtre) et au niveau de la conception des messages.

Chapitre 5. Fonctionnalités et classification des logiciels EDI

D'après ce qui a été dit dans la première partie de cet exposé, on s'attend naturellement à ce que les fonctionnalités d'un logiciel de traduction EDI assurent la transmission et la réception de messages EDI et leur traduction dans des formats appropriés.

En ce qui concerne la classification des logiciels EDI, le rapport TEDIS de la Communauté européenne [TEDIS, 1990] comporte une analyse des besoins des utilisateurs (159 d'entre-eux se sont exprimés) en logiciels EDI. Dans ce rapport, les besoins sont comparés avec les services offerts par une série de 70 logiciels disponibles au moment de la réalisation de cette étude.

En fait, cette étude a été accomplie en deux étapes (TEDIS, 1990, p 6):

La première étape a consisté à:

- évaluation des produits logiciels disponibles sur le marché;
- évaluation de l'expérience des utilisateurs et définition de leurs besoins;
- comparaison des fonctionnalités des produits disponibles et des expériences des utilisateurs et énumération des fonctionnalités requises;

La seconde étape s'est concrétisée par l'établissement de recommandations pour l'amélioration des produits existants et la définition d'une spécification fonctionnelle pour la conception et le développement de logiciels adaptés.

Ce qui nous intéresse plus particulièrement dans ce rapport est une classification des logiciels de traduction EDI en trois grandes catégories, classification basée sur le mode d'utilisation des logiciels et sur leurs fonctionnalités.

Dans un premier temps, nous allons décrire les principaux types de fonctionnalités que l'on retrouve dans les logiciels EDI. Ensuite, nous reprendrons la définition des catégories de logiciels proposée dans le rapport TEDIS.

5.1. Fonctionnalités des logiciels EDI

Globalement, le modèle de base des fonctions d'un logiciel EDI comprend 5 parties (figure 5.1):

- les fonctions d'entrée et de sortie,
- la conversion,
- la communication externe, vers le réseau,
- la communication interne, vers l'application,
- les fonctions de gestion.

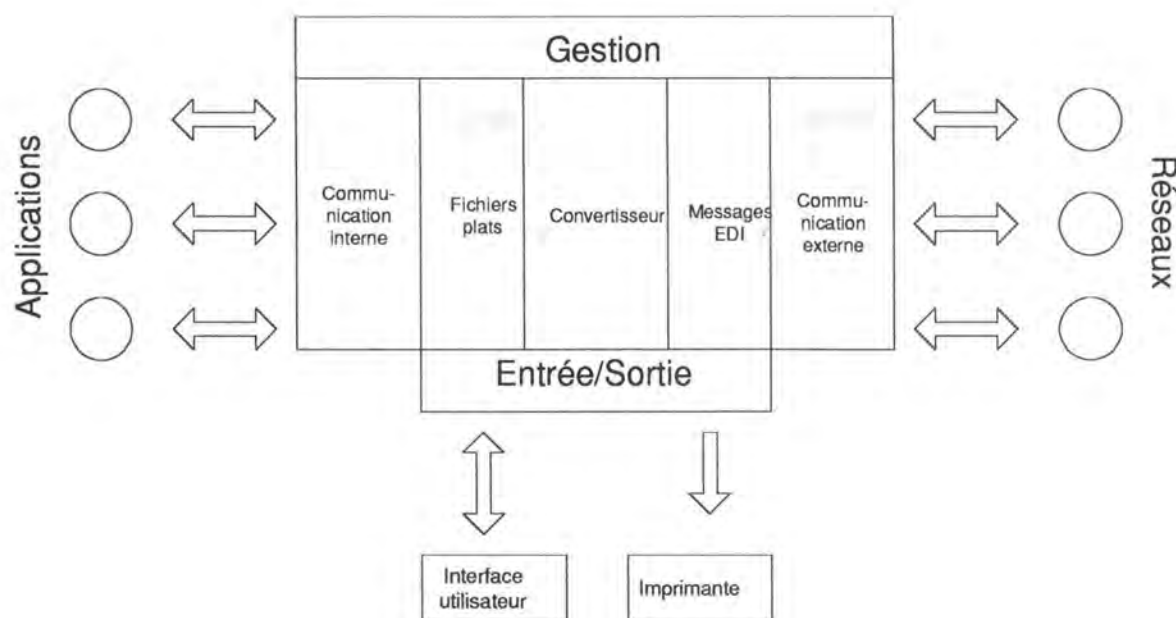


Figure 5.1: Les composants principaux d'un logiciel EDI

5.1.1. Les fonctionnalités d'entrée et sortie

En ce qui concerne les fichiers plats, bon nombre de logiciels EDI offrent la possibilité à l'utilisateur d'introduire ses données au moyen d'un écran de saisie. Il existe plusieurs sortes d'écrans de saisie, réparties en deux grandes catégories:

- écrans dont la présentation des zones de dialogues correspond à la structure des messages EDI, avec une terminologie EDIFACT par exemple. Dans ce cas, le logiciel EDI ne s'occupe que de la vérification de la validité syntaxique des données.

- écrans de saisie de type fonctionnel, dans lesquels les éléments de données sont exprimés en des termes empruntés à la terminologie de l'application et non au standard EDIFACT. Le logiciel se charge donc de représenter les données introduites par l'utilisateur sous une forme correspondant à EDIFACT. Par exemple, le logiciel ajoute lui-même les qualifiants.

Le premier type d'écran ne peut être utilisé que par des utilisateurs expérimentés, car il est proche d'EDIFACT.

Les fonctions de sortie fournissent la possibilité d'imprimer des messages, avec la fonction additionnelle qui consiste à définir les formats d'impression. Dans d'autres cas, les messages ne peuvent être imprimés que dans un format prédéfini ou dans un format EDI.

Par format EDIFACT, nous entendons un format dans lequel les éléments de données sont ordonnés et exprimés conformément à EDIFACT.

5.1.2. La conversion

La partie d'un logiciel EDI qui accomplit les fonctions de conversion est appelée le convertisseur (ou traducteur). Le convertisseur traduit les messages d'un format interne propre à l'application (généralement stockés dans un fichier plat interne) en format EDI et vice versa.

Plus la flexibilité (par rapport à la structure des fichiers de l'application) offerte par le convertisseur est grande, moins importantes sont les modifications à apporter aux applications qui produisent ces fichiers.

Fondamentalement, le processus de traduction recouvre deux aspects:

- la conversion syntaxique,
- la conversion sémantique.

Tout au long de la conversion syntaxique, les messages de format interne sont convertis en messages EDI syntaxiquement corrects. Cela signifie que le résultat obéit au diagramme de structure du message (caractère obligatoire/facultatif, nombre maximum d'occurrences de groupes ou de segments, usage correct des séparateurs, ...) et utilise le bon ensemble de caractères. De plus, le convertisseur peut générer les segments de services ou des valeurs spécifiques d'éléments de données, comme les compteurs de contrôle et les numéros de référence.

La conversion sémantique inclut le traitement correct des qualificants, la conversion des unités (exemple: miles/kilomètres), la conversion minuscules/majuscules, l'ajustement de la taille et de la précision des éléments de données. La conversion des codes appartient aussi à cette catégorie. En effet, une application peut utiliser des codes différents de ceux préconisés au niveau international. Le convertisseur peut automatiquement les convertir en codes standards en utilisant des tables de codes.

Le traitement des qualificants spécifie la fonction d'un élément de donnée ou d'un segment générique. Ainsi, les qualificants peuvent fortement influencer la signification des données dans un segment ou dans un élément de donnée. Normalement, les applications n'ont pas l'usage de qualificants. A partir du contexte ou du type d'un record dans un fichier interne, le convertisseur devrait pouvoir dériver les valeurs de qualificants appropriées et l'ajouter au segment ou à l'élément de donnée correspondant. Au cours du processus de traduction, les qualificants doivent être gérés de façon similaire.

La structure des messages EDI et des fichiers internes est habituellement définie dans des « tables de définition » (definition tables). Un convertisseur fait usage de telles tables pendant le processus de traduction. Un convertisseur peut utiliser des tables séparées pour définir:

- la structure des messages EDIFACT,

- la structure des fichiers internes,
- la conversion des codes.

De plus, la plupart des convertisseurs peuvent produire des rapports d'erreurs en cas de non-respect par les messages traduits des définitions incluses dans ces tables.

5.1.3. La communication interne

Les fonctionnalités de communication interne relient les applications internes au logiciel de traduction. Il peut s'agir du routage (automatique) des messages au travers de différents systèmes internes, probablement sur différentes plates-formes hardware. Généralement, une application interne produit un fichier interne qui est un fichier qui contient un ou plusieurs messages dans un format interne. Les fonctions de communication déchargent ce fichier et le préparent pour un traitement ultérieur, par exemple pour la traduction. Les messages reçus sont stockés dans un endroit particulier, et, si nécessaire, divisés en plusieurs fichiers séparés qui doivent être distribués dans des applications internes séparées.

5.1.4. La communication externe

Cette partie du logiciel forme l'interface au réseau utilisé pour la transmission des interchanges aux partenaires. Habituellement, lorsqu'on implémente l'EDI, on choisit un service de boîte aux lettres électronique. Ce faisant, les interchanges sont stockés dans des endroits connus et peuvent être récupérés à n'importe quel moment.

5.1.5. La gestion des événements

Les fonctionnalités de gestion que les logiciels peuvent offrir sont les suivantes:

- comptes rendus des opérations,
- rapports qui explicitent les erreurs survenues.

Les logiciels EDI utilisent normalement des tables et des fichiers qui décrivent comment les messages doivent être traduits et transmis. Les logiciels peuvent contenir des outils interactifs qui permettent de mettre à jour les caractéristiques contenues dans ces tables et ces fichiers.

5.2. Catégories de logiciels EDI

Conformément au modèle de base présenté ci-dessus (figure 5.1), les produits logiciels EDI peuvent être classés comme suit:

- les convertisseurs EDI,
- les passerelles EDI,
- les stations de travail EDI.

5.2.1. Les convertisseurs EDI

Les convertisseurs EDI sont les outils EDI de base. « Leur seule fonction consiste à construire et à traduire des messages » [TEDIS, 1990, p14]. Un convertisseur convertit les données d'un fichier de format interne, issu d'une application interne, en un message construit conformément au standard EDI. Les messages reçus sont traités dans le sens inverse. La structure des messages EDI est définie dans des tables. Un convertisseur produit aussi des fichiers d'erreurs.

5.2.2. Les passerelles EDI

Par rapport à un convertisseur, une passerelle supporte en plus l'interfaçage avec les applications internes (communication interne) et le réseau (communication externe). Elle relie une ou plusieurs applications internes au réseau externe en gérant:

- plusieurs partenaires,
- des réseaux multiples,
- des applications multiples.

Une passerelle utilise souvent des routines qui accomplissent le transfert des données en direction et en provenance des applications internes. En réalité, une passerelle fait appel au convertisseur avec le fichier interne comme paramètre d'entrée. Elle peut également supporter une variété de logiciels de communication et générer automatiquement des accusés de réception.

La caractéristique la plus marquante d'une passerelle est sa capacité à traiter des messages sans aucune intervention humaine. Dans ce cas, la capacité à produire des rapports d'état prend toute son importance.

5.2.3. Les stations de travail EDI

Une station de travail EDI est une passerelle fournie avec une option « data-entry » manuelle. En utilisant une station de travail, un utilisateur final peut traiter les messages manuellement. Ici, la convivialité est évidemment une des caractéristiques principales. Une station de travail ne comprend pas nécessairement l'interface aux applications internes, mais les auteurs du rapport TEDIS ont fait le choix de définir une station de travail comme « un outil EDI contenant aussi toutes les fonctionnalités d'une passerelle ».

Il existe deux types de stations de travail:

- des stations de travail génériques, qui sont capables de traiter tous les types de messages;
- des stations de travail spécifiques, capables de traiter un ensemble restreint de messages utilisés au sein d'une application spécifique.

A présent, nous connaissons les principales caractéristiques d'un logiciel EDI. Une étape cruciale dans la mise en oeuvre d'une relation EDI consiste à choisir un logiciel EDI. Et pour réaliser un choix, il faut disposer d'une série de critères. C'est ce que nous allons voir dans le chapitre suivant.

Chapitre 6. Choix d'un logiciel EDI

Une étude a été réalisée [SORTLAND, 1990] au sujet des critères à considérer pour choisir un logiciel EDI que l'on désire installer sur un micro-ordinateur. Cette situation est typiquement représentative du contexte dans lequel nous avons travaillé.

Dans ce contexte, l'auteur distingue deux modes d'utilisation:

- le mode « stand-alone » dans lequel le micro-ordinateur réalise toutes les opérations sur les données et gère lui-même le stockage des données,
- le mode « front-end », dans lequel le micro-ordinateur sert de passerelle entre le monde extérieur et l'application (où se trouve l'information à échanger) gérée par un système séparé.

Il faut noter que certaines fonctions de base des logiciels EDI sont communes aux deux modes d'utilisation:

- l'information contenue dans l'application doit être traduite de façon à obtenir des fichiers dont le format satisfait au standard EDI utilisé;
- tout fichier à transmettre qui contient des données dans un format EDI doit pouvoir être transmis au moyen d'un réseau approprié;
- les fichiers de format EDI reçus des partenaires doivent pouvoir être traduits sous un format utilisable par l'application.

Ces fonctions de base doivent être complétées par des fonctions de maintenance des fichiers qui contiennent l'information nécessaire aux échanges avec les partenaires.

Dans le mode « stand-alone », des écrans de saisie de données et des fonctionnalités d'impression de documents doivent être proposés par le logiciel à l'opérateur humain qui le manipule. Donc, la sélection d'un logiciel doit tenir compte de ces caractéristiques ergonomiques.

Par contre, pour un environnement « front-end », il n'est pas nécessaire de rechercher un logiciel qui offre de telles caractéristiques. En effet, c'est le système hôte de l'application qui doit être convivial. Les critères importants pour les utilisateurs de tels systèmes sont:

- la clarté et la facilité d'utilisation des fonctionnalités qui permettent à l'utilisateur de spécifier la relation entre le format des fichiers de l'application et le format des fichiers EDI,
- la possibilité de transférer facilement des fichiers entre le système hôte de l'application et le logiciel installé sur micro-ordinateur,
- les moyens offerts à l'utilisateur pour automatiser le fonctionnement de la passerelle EDI.

L'auteur préfère utiliser le terme « logiciel de gestion EDI » (EDI Management Software) plutôt que le terme « logiciel de traduction EDI » (EDI Translation Software) car l'ensemble des fonctionnalités destinées à établir des relations EDI dans une organisation est plus complexe que celui nécessaire à la traduction de fichiers. En effet, il existe un monde de différences entre le fait d'effectuer de temps en temps une traduction de messages avec un partenaire et le fait d'être impliqué dans des relations intensives avec beaucoup de partenaires et de messages différents. Dans ce dernier cas, les échanges concernent une multitude de données, impliquent souvent des réseaux différents à gérer et la nécessité de planifier les activités et de les automatiser au maximum.

Ainsi, l'acheteur potentiel d'un logiciel EDI doit savoir quels sont, et quels seront, ses besoins d'échanges et leur complexité.

Comme le fait remarquer l'auteur, le prix d'achat d'un logiciel ne représente qu'une faible part des coûts associés à la mise en oeuvre d'une relation EDI (formation des utilisateurs, programmation des applications, ...)

L'objectif de l'étude [SORTLAND, 1990] est d'identifier les paramètres qui sont à considérer lorsque l'on choisit un système basé sur un environnement micro-ordinateur pour supporter les besoins en EDI. En fait, il émet des suggestions concernant l'importance relative des critères qu'il présente dans la démarche d'évaluation. Il se base sur son expérience de plusieurs années dans le domaine du conseil en systèmes EDI adaptés à des utilisateurs variés.

Le processus de sélection d'un système EDI doit commencer par une analyse de l'environnement de l'utilisateur. C'est seulement lorsque cette étape est accomplie que l'on peut comparer et donner un poids aux critères d'évaluation. Par exemple, les caractéristiques de l'interface utilisateur (écrans) plus d'importance dans le cas d'un système stand-alone que dans un environnement front-end.

Les critères de sélection sont regroupés en dix sections. Chaque section comprend un commentaire sur l'impact potentiel qu'elle représente dans le choix d'un système EDI. Un facteur de pondération (variant de 1 à 5 pour exprimer l'importance croissante des critères) est assigné aux critères.

6.1. Groupe 1: Informations concernant la société qui fournit le logiciel

Il est important de bien connaître les fournisseurs potentiels du logiciel que l'on désire se procurer. En particulier, comme le monde de l'EDI évolue rapidement (changements dans les relations avec les partenaires et évolution des environnements informatiques), il est essentiel de pouvoir compter sur les capacités du fournisseur à s'adapter au marché. Celui-ci doit en outre garantir un service d'assistance et de conseil efficace.

Les informations à recueillir sont notamment les suivantes:

- l'historique de la société qui fournit le logiciel,
- le degré d'importance de l'EDI parmi les activités de la société,
- la réputation du produit potentiellement intéressant,
- le rythme auquel le produit subit des modifications (améliorations) et les améliorations prévues.

L'auteur attribue un poids compris entre 2 et 4 aux critères de ce groupe. Un poids de 2 est particulièrement adapté aux environnements les moins complexes. Cela signifie donc que ce premier groupe de critères n'est pas d'une importance cruciale dans ces types d'environnements.

6.2. Groupe 2: L'interaction entre l'utilisateur et l'interface (Screen functionality)

Ce groupe est de première importance pour les utilisateurs de systèmes stand-alone, amenés à effectuer toutes les opérations nécessaires à la mise en oeuvre d'échanges de messages uniquement à partir de leur PC. La convivialité de l'environnement logiciel est par conséquent primordiale.

Parmi les critères à considérer, on peut citer les suivants:

- une conception ergonomique des relations avec l'utilisateur, par l'intermédiaire d'écrans conviviaux (messages d'erreurs, aide, ...)
- la possibilité d'éditer les fichiers de données que l'on désire échanger.

En vertu de ce qui vient d'être dit, il n'est pas étonnant que l'auteur attribue un poids de 4 à 5 à ce groupe dans le cadre d'un environnement de type stand-alone. Pour un environnement de type front-end, le poids n'est que de 1.

6.3. Groupe 3: Les fonctionnalités offertes par le système

Les critères englobés dans cette section sont relatifs aux fonctionnalités liées à l'exécution des processus de traitement des messages (les processus de traduction), à savoir:

- l'archivage des fichiers,
- la détection et la gestion des erreurs,
- les possibilités de création de rapports statistiques et de rapports sur le déroulement des processus,
- le suivi des fichiers échangés,

Comme ces caractéristiques sont importantes dans le cas de toutes les installations, un poids de 5 leur est attribué. La détection et la gestion des erreurs dans les données entrantes sont très importantes pour les utilisateurs du système front-end

car, comme les processus sont automatisés, les erreurs risquent de passer inaperçues et de se retrouver dans l'application.

6.4. Groupe 4: La flexibilité

Ce groupe concerne les caractéristiques que contrôle l'utilisateur, comme:

- le fait de pouvoir intégrer des fonctionnalités conçues par l'utilisateur,
- la capacité de fixer des paramètres de traduction spécifiques pour chaque partenaire, et de leur associer certains types de messages,
- la possibilité de construire et de mettre à jour les tables de paramètres de mapping,
- la capacité de configurer le système selon les besoins spécifiques de l'utilisateur,
- la modularité de la conception du logiciel de sorte que l'ajout de modules améliorés puisse être réalisé de façon incrémentale et aisée sans affecter les autres composants du logiciel.

Excepté le cas de relations EDI très limitées (un seul partenaire par exemple), la flexibilité est essentielle et indépendante des modes de fonctionnement (front-end et stand-alone). Le poids attribué au groupe est égal à 5.

6.5. Groupe 5: Les standards EDI

Ce groupe concerne la diversité des standards supportés par le logiciel et la manière dont le logiciel exploite la connaissance de ces standards:

- le logiciel supporte-t-il l'entière d'un standard ou seulement un sous-ensemble utilisé dans un domaine bien précis?
- le contrôle de la validité des données est-il complet, et à quel niveau (segments et éléments de données) ? Concerne-t'il aussi bien les données entrantes que les données sortantes?
- l'utilisateur peut-il définir le degré de contrôle des données?
- que se passe-t-il en cas d'erreur lors du contrôle? Est-ce que cela satisfait l'utilisateur?
- existe-t-il plusieurs versions des standards? Si oui, comment peut-on les spécifier?
- comment s'effectue la modification de la liste des messages à échanger?

Le système est d'autant plus intéressant qu'une réponse positive est apportée aux questions ci-dessus. Néanmoins, l'auteur considère que la possibilité éventuellement laissée à l'utilisateur de contrôler la sévérité des contrôles syntaxiques n'est d'aucune utilité. Il est en effet préférable de conserver un contrôle maximal: si un utilisateur est moins strict, les données qu'un partenaire reçoit de lui risquent d'être rejetées. A moins d'imposer certaines règles qui consistent en un amendement du standard, ce qui n'est pas désirable. Par conséquent, le poids attribué au groupe est fixé à 3 ou 4.

6.6. Groupe 6: Les comptes rendus des opérations effectuées (reporting)

Afin de rendre l'utilisation d'un logiciel EDI plus intéressante, il est bon de pouvoir disposer de comptes rendus des opérations effectuées. Ces rapports doivent être adaptés aux besoins de l'utilisateur du logiciel:

- des rapports de contrôle (erreurs, déroulement des opérations, ...) sont-ils disponibles, fournissant à l'utilisateur des informations intéressantes concernant le statut des opérations effectuées?
- peut-on imprimer dans un format lisible les messages échangés?
- l'utilisateur peut-il modifier les rapports?
- est-ce que les fichiers de données sources des rapports sont disponibles pour que l'utilisateur puisse définir des rapports personnalisés?

L'auteur n'accorde qu'un poids moyen de 2 à l'ensemble de ces critères. Il juge les autres groupes beaucoup plus importants.

6.7. Groupe 7: La communication

L'aspect communication est évidemment un aspect essentiel de l'EDI. Aussi faut-il veiller aux critères suivants:

- quels sont les services de communication supportés par le logiciel?
- la communication est-elle intégrée aux processus de traduction?
- en cas d'erreur de transmission, existe-t-il une information accessible à l'utilisateur pour qu'il soit prévenu?

Il est préférable que les fonctionnalités liées à la communication soient intégrées aux fonctionnalités de traduction des messages. En fait, l'importance de ces critères est à pondérer en fonction des exigences des relations avec les partenaires.

6.8. Groupe 8: Installation et maintenance

Par installation, l'auteur entend la mise en oeuvre du logiciel de façon à ce qu'il soit opérationnel:

- le logiciel peut-il être installé facilement, sans devoir consulter le fournisseur?
- La documentation est-elle adéquate?
- y a-t-il des menus pour la configuration du système et l'établissement des caractéristiques des relations avec les partenaires? Sont-ils faciles à utiliser?
- y a-t-il une formation?
- existe-t-il un tutoriel?
- comment sont fournies les mises à jour des standards?

Au cours du temps, la maintenance est certainement plus importante que l'installation. Le facteur de pondération des critères de ce groupe est fixé à 3.

6.9. Groupe 9: Le support

Ce point est très important, comme nous l'avons mentionné plus haut:

- est-ce qu'un support à l'installation est prévu, s'il est nécessaire?
- existe-t-il une « hot line » pour bénéficier d'une assistance téléphonique?
- quelle est la qualité du manuel de l'utilisateur et des autres documents éventuels?

Il est important d'être sûr que le logiciel que l'on choisit bénéficie d'un support au moment de l'achat et après l'achat. Par conséquent, le poids qui est attribué à ce critère est égal à 5.

6.10. Groupe 10: Les coûts

Le prix d'achat d'un logiciel EDI ne représente qu'une partie des coûts d'implémentation. Il faut en effet prendre en compte le coût de la programmation des applications et des caractéristiques nécessaires à la réalisation des relations commerciales au moyen de l'EDI.

- quel est le coût du logiciel et des opérations à réaliser pour l'intégrer?
- existe-t-il des produits standards qui peuvent être intégrés tels quels dans l'environnement de l'utilisateur?
- une maintenance annuelle est-elle prévue et à quel prix?
- le logiciel peut-il être prêté (loué) pour une période d'essai?

Comme le coût du logiciel est faible par rapport à l'investissement total, son importance est faible, tout comme son poids fixé à 1.

En conclusion, les questions qu'il est le plus important de se poser sont:

- l'utilisateur est-il satisfait du fournisseur en termes de support, actuellement et à l'avenir?
- le produit répond-t-il bien aux besoins de l'utilisateur?
- le coût total est-il raisonnable?

A présent, nous disposons de suffisamment d'informations sur la classification et les caractéristiques générales des logiciels EDI pour pouvoir présenter celui que nous avons choisi, ATLAS EDI.

Dans ce qui suit, nous présentons le logiciel ATLAS EDI, sur base des critères cités dans [SORTLAND, 1990] et dans le rapport TEDIS.

Chapitre 7. Présentation d'ATLAS EDI

Le logiciel ATLAS EDI est installé sur un PC, conformément au mode stand-alone décrit plus haut [SORTLAND, 1990]. Ainsi, il réalise toutes les opérations sur les données et gère lui-même leur stockage. Il correspond aussi à une station de travail EDI selon la terminologie utilisée dans le rapport TEDIS.

On distingue trois parties dans le logiciel:

- l'administration du système (System Administration),
- le traitement des données (Data Processing),
- la gestion des communications (Communications).

L'administration du système est la partie la plus importante puisqu'elle permet d'effectuer:

- l'enregistrement des renseignements relatifs à la société de l'utilisateur,
- l'enregistrement du profil du réseau par lequel les messages vont être acheminés,
- la définition du profil des partenaires commerciaux,
- la définition des paramètres utilisés par les processus de traduction,
- la définition de data maps (data mapping),
- la possibilité de faire appel à un job scheduler qui permet de regrouper une série de tâches à exécuter dans un même job qui peut être exécuté quand l'utilisateur le désire.
- la paramétrisation du système (couleurs de l'écran, mode de liaison vers l'imprimante...)

Les caractéristiques techniques d'ATLAS EDI sont décrites selon le canevas présenté ci-dessus [SORTLAND, 1990]. Conformément à ce canevas, nous commentons chaque groupe de critères à l'aide des caractéristiques propres à ATLAS EDI.

7.1. Informations concernant la société qui fournit le logiciel

En réalité, nous ne disposons pas de beaucoup d'informations au sujet de l'historique de la société qui produit le logiciel (ATLAS Products International Limited). Par contre, nous bénéficions de l'assistance de la société OFFIS S.A. qui distribue ce produit en Belgique et sur laquelle nous pouvons compter, par l'intermédiaire de Monsieur Pascal Waterkeyn. De plus, la réputation du produit n'est plus à faire.

7.2. Interactions entre l'utilisateur et l'interface (Screen functionality)

En règle générale, nous pouvons dire qu'ATLAS EDI permet à l'utilisateur d'utiliser beaucoup de fonctionnalités et d'obtenir des informations sur les opérations réalisées.

Malheureusement, l'interface souffre du fait que le logiciel est géré par le DOS et ne bénéficie pas de la convivialité fenêtrée de Windows. Un autre point négatif est la pauvreté de l'aide en ligne offerte à l'utilisateur. Celle-ci renvoie bien (trop) souvent l'utilisateur au manuel, qui n'est pas vraiment un exemple de clarté.

7.3. Fonctionnalités offertes par le système

Pour rappel (cf. paragraphe 6.3.), les critères englobés dans cette section par [SORTLAND, 1990] sont relatifs aux fonctionnalités associées aux processus de traduction. Il s'agit des fonctionnalités:

- d'archivage,
- de détection et de gestion des erreurs,
- de création de rapports statistiques et de rapports sur le déroulement des processus,
- de suivi (tracking) des fichiers échangés.

Nous commencerons par une description des processus de traduction, ce qui nous permettra d'introduire progressivement le rôle joué par les fonctionnalités qui leur sont associées. La figure 7.1 nous donne une représentation simplifiée de l'architecture et du fonctionnement d'ATLAS-EDI. Nous avons personnellement déduit cette architecture sur base du fonctionnement du logiciel et de la structure de ses fichiers. Ce schéma nous servira de référence pour situer chaque fonctionnalité au sein du fonctionnement global du logiciel ATLAS.

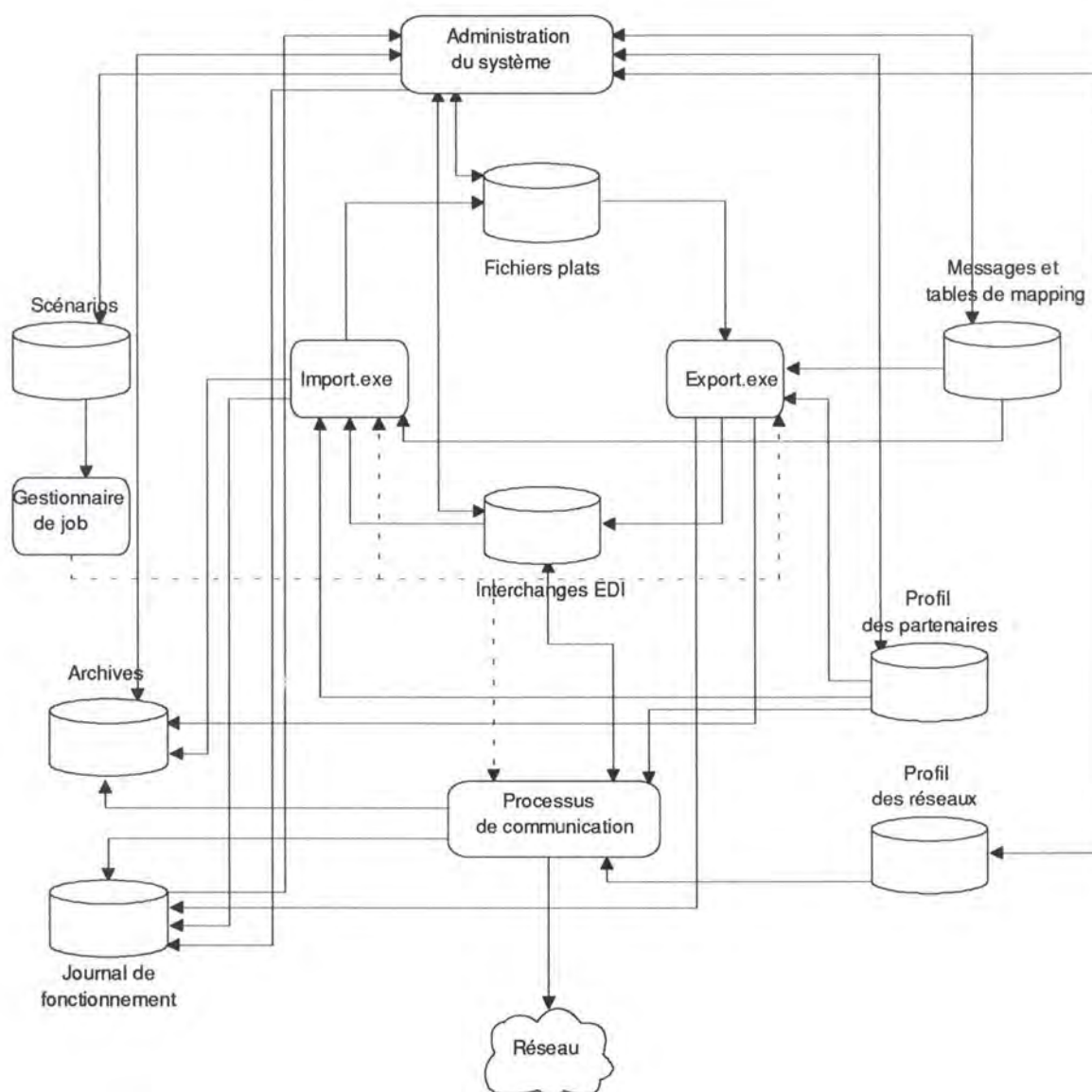


Figure 7.1: représentation simplifiée de l'architecture et du fonctionnement d'ATLAS-EDI

Il existe deux types de processus de traduction :

-la traduction d'un fichier plat en un (ou plusieurs) fichier(s) EDI destiné(s) à être envoyé(s) à un (des) partenaire(s). Ce type de processus porte le nom d'EXPORT, et nous dirons qu'un fichier EDI obtenu de cette manière est un fichier exporté.

-la traduction (IMPORT) d'un fichier EDI reçu du partenaire et destiné à être traduit en un (des) fichier(s) plat(s) pouvant être utilisé(s) par une application interne. Nous dirons qu'un fichier plat obtenu de cette façon est un fichier importé. Dans la suite, et pour simplifier l'exposé, nous considérons qu'un fichier plat ne donne naissance qu'à un seul fichier EDI et réciproquement qu'un fichier EDI ne donne naissance qu'à un seul fichier plat. Cependant, il s'agit là d'un cas particulier.

La traduction EXPORT est donc le processus de conversion d'un fichier plat d'une application en un fichier de format EDI. Dans le cadre d'une intégration complète avec une application, ce processus de traduction est invoqué après que l'application a créé le fichier plat. Au sein d'ATLAS, les fichiers plats prêts à être exportés doivent suivre la convention d'appellation: e-<caractères>.<caractères>. En cas de succès du processus de traduction, les données traduites sont stockées dans un fichier dont le nom obéit à la convention suivante: s-<méthode de transfert>.nnn où « nnn » est un numéro de génération attribué par le processus de traduction au fichier nouvellement créé (cf. figure 7.2). Tous les fichiers créés lors d'un même processus d'export porteront le même numéro de génération. Ce numéro est incrémenté d'une valeur de 1 à chaque nouvelle exécution du processus EXPORT. Quant au fichier qui vient d'être traduit, il est renommé suivant le format e-<caractères>.nnn et archivé. La fonctionnalité d'archivage d'ATLAS-EDI permet la sauvegarde (dans un répertoire particulier) d'une copie de tout fichier subissant un processus de traduction ou d'expédition sur le réseau. Si l'un de ces processus doit être recommencé, il pourra l'être sur base de cette copie du fichier initial. Cet archivage est déclenché automatiquement par les processus de traduction (import.exe et export.exe) ou de communication (cf. figure 7.1). Le système d'administration d'ATLAS permet la manipulation de ces archives et leur réincorporation au sein des répertoires de travail des processus concernés.

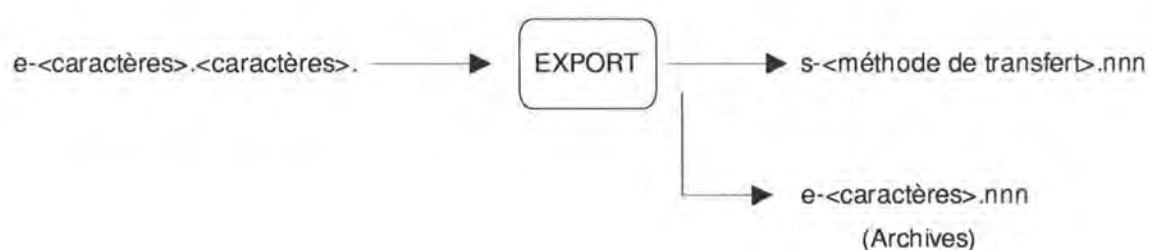


Figure 7.2: processus de traduction EXPORT sans erreur

Lorsque la traduction s'effectue, l'écran contient une fenêtre donnant une série d'informations sur l'avancement du processus de traduction. Parmi ces informations, on trouve notamment le nombre d'erreurs rencontrées. Deux catégories d'erreurs peuvent apparaître :

- les erreurs (dans les types de données, dans leur taille,...) qui n'interrompent pas la traduction,
- les erreurs qui provoquent l'arrêt de la traduction (« fatal errors »), comme par exemple l'absence de la référence du partenaire dans le fichier plat.

Quoiqu'il arrive, la routine de traduction s'achève toujours en indiquant un niveau d'erreur:

- niveau 0: la traduction est complètement réussie,
- niveau 1: une erreur mineure est survenue, telle qu'une taille de champ incorrecte,
- niveau 2: le fichier n'a pu être traduit, à cause d'une erreur qui a provoqué l'interruption de la traduction (comme par exemple une séquence de segments invalide),
- niveau 3: une erreur de système est apparue telle que l'impossibilité d'écrire sur le disque dur ou signale un manque de mémoire.

Si le niveau d'erreur est non nul, il est recopié dans un fichier appelé e-errnum.

Si la traduction a dû être interrompue, la partie du fichier plat qui n'a pu être traduite est recopiée dans un fichier appelé e_errors.nnn (cf. fig 7.3) où « nnn » est le numéro de génération associé par le processus au fichier créé.

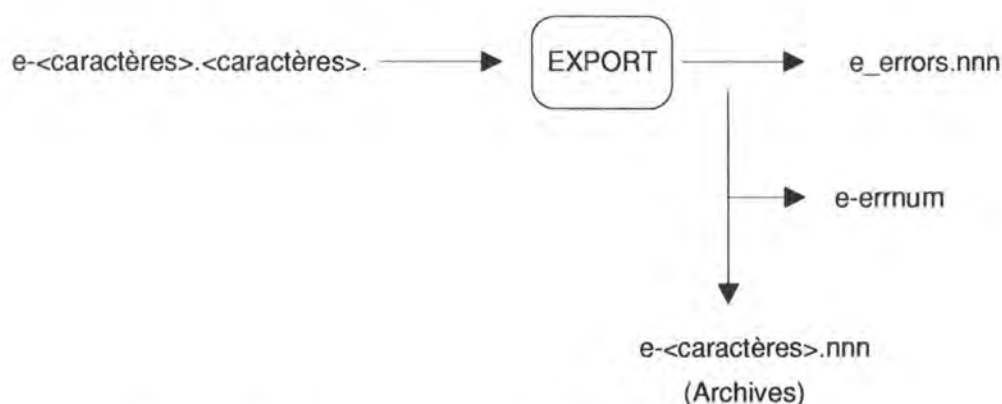


Figure 7.3: processus de traduction EXPORT avec erreur(s)

Si l'utilisateur désire connaître les messages d'erreurs générés, il doit consulter le rapport de fonctionnement relatif au processus de traduction qui vient d'être réalisé. En fait, l'ensemble des opérations effectuées lors de chaque session ATLAS (une session ATLAS débute lors du lancement du logiciel et se termine lorsqu'on le quitte) est enregistré dans ce que l'on pourrait appeler le journal de fonctionnement d'ATLAS (Audit Trail Reports en anglais), (cf. figure 7.1). Celui-ci se charge également d'enregistrer différents messages liés à la détection d'une erreur lors d'un processus de traduction.

La traduction **IMPORT** est le processus de conversion de fichiers de format EDI en fichiers plats destinés à être utilisés par des applications.

Cette traduction est invoquée après la réception de données EDI en provenance le plus souvent d'un réseau. Les fichiers EDI doivent suivre la convention de dénomination suivante: `r-<méthode de transfert>.nnn` où « méthode de transfert » est le nom de la méthode de transfert utilisée pour obtenir le fichier. Le suffixe « nnn » est le numéro de génération géré par le processus de communication utilisé

pour réceptionner le fichier EDI (les processus EXPORT et de communication gèrent chacun leurs propres numéros de génération). Ce numéro de génération sera commun à tous les fichiers issus du processus de traduction IMPORT portant sur un même fichier EDI.

Si la traduction se passe correctement (cf. figure 7.4), on obtient:

- un fichier plat destiné à être utilisé par une application: i-<nom document>.nnn
- un fichier archivé, r-<méthode de transfert>.nnn, qui est une copie du fichier EDI traduit.

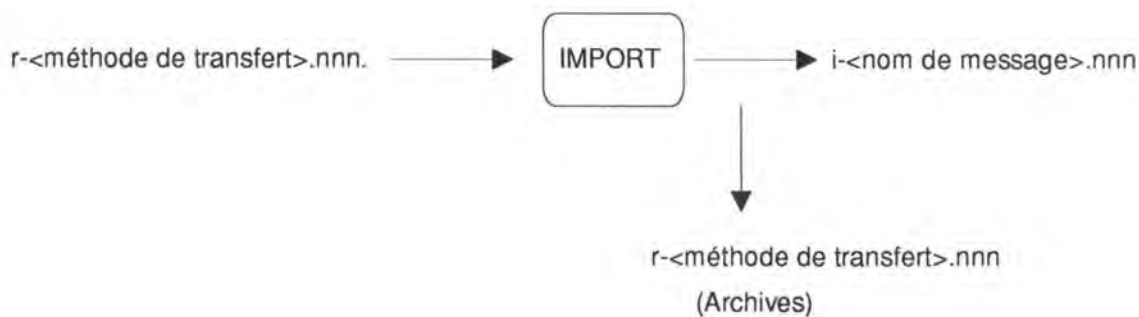


Figure 7.4: processus de traduction IMPORT sans erreur

En cas d'erreurs, le niveau d'erreur est recopié dans un fichier appelé i-errnum. Quant au fichier EDI incorrect, il est recopié dans un fichier de type r-errors.nnn (cf. figure 7.5).

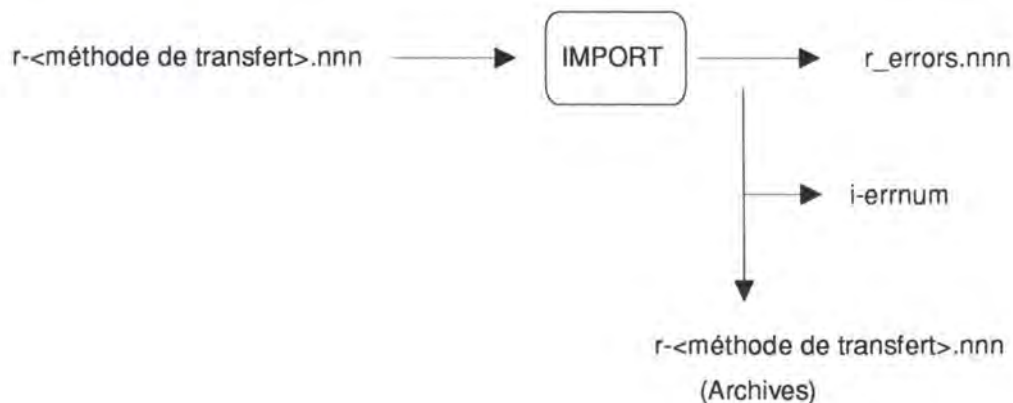


Figure 7.5: processus de traduction IMPORT avec erreur(s)

Comme dans le cas d'un processus EXPORT, si l'utilisateur désire connaître les messages d'erreurs générés il doit consulter le rapport de fonctionnement relatif au processus de traduction qui vient d'être réalisé.

ATLAS gère également un système de suivi (tracking) des fichiers traduits et/ou échangés avec des partenaires (système non représenté sur la figure 7.1).

Nous terminons cette description fonctionnelle d'ATLAS par la présentation du gestionnaire de job (cf. figure 7.1). Celui-ci permet de gérer une série de scénarios (job) d'enchaînement de différentes fonctions d'ATLAS. On peut par exemple définir un job permettant de faire suivre une traduction d'un fichier plat de son expédition via un réseau de communication.

Notons pour terminer que les processus de traduction EXPORT et IMPORT peuvent être aussi bien invoqués à l'aide d'un menu du logiciel qu'à partir de l'invite du DOS (et donc sans devoir lancer le programme ATLAS complet). Dans ce dernier cas, les noms des fichiers à traduire sont ajoutés comme paramètres de la commande de traduction (EXPORT <nom de fichier1> <nom de fichier2>...).

7.4. La flexibilité du logiciel

Dans ce groupe de caractéristiques, on retrouve notamment les suivantes:

- la capacité de spécifier des paramètres de traduction différents pour chaque partenaire et d'associer à un partenaire certains types de messages,
- la possibilité de construire et de mettre à jour les tables de mapping (une table de mapping est une table établissant la correspondance entre les champs de données d'un fichier plat et ceux du fichier EDI correspondant, et ce, pour chaque sens de traduction),
- la capacité de configurer le système selon les besoins spécifiques de l'utilisateur.

7.4.1. Spécification des paramètres de traduction et association de certains types de messages à chaque partenaire

Les profils de partenaires sont de deux types: le profil de l'utilisateur et celui de ses partenaires.

Le profil de l'utilisateur reprend des renseignements généraux à propos de l'utilisateur. Une fois ces informations enregistrées, elles peuvent être automatiquement sélectionnées pour être incorporées dans les fichiers EDI que l'on désire envoyer. Ainsi, les informations que l'on renseigne dans cette section ne doivent pas nécessairement figurer dans les fichiers plats de l'application. Ils sont toujours valables et peuvent être référencés dans tous les cas. Notons bien qu'il s'agit là d'une possibilité offerte à l'utilisateur, et non d'une obligation. Nous reviendrons sur ce point plus tard lorsque nous détaillerons la construction des tables de mapping.

Les informations relatives à un partenaire commercial sont de première importance et dépassent la simple connaissance de son nom et de son adresse. Lors d'une traduction EXPORT d'un fichier plat par exemple, le logiciel commence par lire dans celui-ci la référence du partenaire. Sur base de cette référence, il peut accéder aux informations qu'il possède sur ce partenaire dans le fichier de profil des partenaires (cf. figure 7.1). Le profil d'un partenaire comprend notamment une liste des types de messages que ce partenaire est susceptible d'échanger. Chaque élément

de cette liste permet au logiciel d'identifier un type de message, son standard, sa syntaxe ainsi qu'une table de conversion (data map) qui devra être utilisée par le processus. Le profil d'un partenaire reprend également le type de réseau utilisé pour communiquer avec ce partenaire, ce qui permettra de déterminer le nom du fichier en sortie du processus EXPORT (cf. 7.3).

Chaque partenaire est décrit par l'ensemble des informations suivantes:

- sa référence (dans le système de l'utilisateur, chaque partenaire est référencé par un code, par exemple par son numéro de client ou de fournisseur);
- le nom de sa société;
- son adresse pouvant être constituée de quatre lignes;
- le code postal de sa société;
- son numéro de téléphone;
- un numéro de référence de dernier interchange envoyé (chaque interchange envoyé à un partenaire commercial est identifié par un Interchange Reference Number unique calculé automatiquement);
- un numéro de référence de dernier interchange reçu (ce champ est mis à jour automatiquement);
- un identifiant propre au partenaire (Traders Own Id.): il s'agit de l'identifiant que le partenaire commercial s'attribue et qu'il place toujours dans le champ « SENDERS ID » du header de l'interchange;
- quatre champs de données: ces champs peuvent être utilisés pour enregistrer n'importe quel type d'information au sujet du partenaire commercial;
- la méthode utilisée pour échanger les messages avec le partenaire: il s'agit d'un des réseaux auxquels le logiciel peut se connecter automatiquement, moyennant l'enregistrement des informations (qui permettent d'identifier le partenaire parmi tous les utilisateurs de ce réseau) nécessaires à l'établissement de la connexion.

Pour être complet, le profil d'un partenaire commercial doit contenir l'ensemble des types de messages que ce partenaire peut échanger avec l'utilisateur. Cette étape vient naturellement après la description de l'identité et les caractéristiques du partenaire.

Il va de soi que l'on peut supprimer un type de message de la liste de ceux affectés à un partenaire pour éventuellement en insérer un autre.

7.4.2. Construction des tables de mapping

L'établissement d'une table de mapping sera largement commenté plus loin (chapitre 8). Comme nous l'avons vu, il est nécessaire de définir une table de mapping afin de pouvoir traduire un fichier plat en un fichier de format EDI et vice versa. Pour ce faire, il existe des commandes qui facilitent la tâche de l'utilisateur dans la définition d'une table de mapping.

Une table de mapping d'ATLAS contient les informations suivantes:

- l'endroit où se trouvent dans le fichier plat les informations à mentionner dans le fichier EDI,
- la position des champs dans le fichier plat où les données extraites du fichier EDI doivent figurer.

Nous y reviendrons plus largement dans le chapitre suivant.

7.4.3. Configuration du système

Ce qui est crucial dans la configuration du système, c'est le choix des paramètres qui interviennent dans le processus de traduction lui-même.

En effet, le logiciel est capable de traduire beaucoup de formats de fichiers en presque n'importe quel standard EDI. Il est donc indispensable que l'utilisateur définisse correctement le format des fichiers plats de son application afin qu'ATLAS-EDI puisse correctement les interpréter au moment de la traduction, aussi bien dans le cas de la traduction IMPORT que dans le cas de la traduction EXPORT. L'utilisateur peut définir la structure des fichiers plats produits par le logiciel au cours de la traduction IMPORT. L'utilisateur peut également définir la structure des fichiers plats que son application produit afin que le logiciel les traduise sous un format EDI.

Quant à la sécurité, le système du logiciel est destiné à restreindre à certains utilisateurs l'accès à différentes fonctions du logiciel. Les utilisateurs peuvent se voir attribuer leur propre identificateur et leur mot de passe afin d'accéder à certaines fonctionnalités du logiciel. Ainsi, on peut par exemple restreindre la définition des caractéristiques des partenaires à une certaine catégorie d'utilisateurs. Ainsi, lorsqu'un utilisateur tape la commande « ATLAS », le système lui présente une boîte d'identification dans laquelle l'utilisateur doit mentionner son identifiant et son mot de passe. Trois essais infructueux successifs provoquent la désactivation d'ATLAS-EDI. Enfin, une tentative d'accès illicite est enregistrée dans un rapport.

Notons que le logiciel autorise l'utilisateur à procéder à des réglages des couleurs de l'écran et à la configuration de l'imprimante qui peut être reliée au système. Nous ne commenterons pas ici ce genre de détails techniques.

7.5. Les standards EDI

Le logiciel supporte entièrement plusieurs standards EDI, et pas seulement des sous-ensemble définis dans un domaine bien précis. Dans notre cas, le standard qui nous intéresse est EDIFACT. Pour celui-ci, ATLAS supporte les répertoires des éléments de données et des segments pour les versions 90.1 à 93.1.

Pendant la traduction, le contrôle de la validité des données peut être modulé par l'utilisateur au niveau des segments et des éléments de données. Ces contrôles s'appliquent aussi bien aux données entrantes qu'aux données sortantes.

La définition des types de messages EDIFACT que l'on désire échanger s'effectue facilement en donnant les renseignements suivants:

- la composition en segments des types de messages,
- le niveau hiérarchique des segments et leur nombre de répétitions éventuelles,
- le statut des segments (obligatoires ou facultatifs).

7.6. Comptes rendus des opérations effectuées (reporting)

Rappelons que l'ensemble des opérations effectuées lors de chaque session ATLAS est enregistré dans ce que l'on pourrait appeler le journal de fonctionnement d'ATLAS (Audit Trail Reports en anglais). En ce qui concerne plus précisément les rapports liés aux processus de traduction, nous aurons l'occasion d'en reparler plus tard lorsque nous détaillerons la configuration du logiciel.

Les rapports ne sont contenus que dans un fichier en accès lecture et ne peuvent être modifiés.

7.7. La communication

Le logiciel offre un service de connexion à plusieurs réseaux qui peut être lié aux processus de traduction (via la définition d'un job).

L'activation d'une session de communication appelle un programme externe qui se charge d'effectuer la communication (expédition et réception des fichiers appropriés dans un ordre correct). La convention de dénomination de ces drivers de communication est la suivante:

c-<nom du réseau>.BAT dans un environnement PC.

7.8. Installation et maintenance

Le logiciel peut être installé très facilement sans devoir consulter le fournisseur. Il suffit d'exécuter le programme d'installation qui se trouve sur l'une des deux disquettes d'installation.

Une formation est proposée par la société API.

Les mises à jour des standards EDI (ainsi que des tables de mapping) peuvent être obtenues sur simple demande.

Malheureusement, il n'existe pas de tutoriel et la documentation livrée avec le logiciel ne permet pas à l'utilisateur novice de comprendre la procédure à suivre la

première fois qu'il entreprend la configuration du logiciel. La documentation n'explique que les fonctionnalités séparément, sans proposer une méthode globale.

7.9. Le support

Une assistance existe pour l'installation du logiciel et sa configuration. Ce peut être utile dans des environnements plus complexes que le nôtre.

Comme les principales fonctions d'ATLAS EDI ont été présentées, nous allons détailler celles qui interviennent dans la configuration du logiciel. Au terme de cette présentation, nous mettrons en évidence les choix qu'il a fallu effectuer pour mener à bien cette opération.

Chapitre 8. Configuration du logiciel

8.1. Introduction

La configuration du logiciel est l'opération fondamentale à effectuer dans le processus de mise en oeuvre d'une relation EDI. Cette configuration se fera sur base de la définition du type de message à échanger avec Craig Parker. Nous reviendrons plus tard (chapitre 12) sur la méthode à suivre pour obtenir la définition d'un type de message, définition qui doit être partagée par les partenaires qui se l'échangent.

Pour le moment, nous nous limitons volontairement à un exemple très simple afin d'illustrer la procédure à suivre pour:

- traduire sous la forme d'un fichier plat (dont le format est défini par notre application) un fichier EDIFACT (version 90.1) émis par notre partenaire;
- traduire sous la forme d'un fichier EDIFACT (dont le format est défini par la version du standard utilisée) un fichier plat de format défini par nous.

Le type de message à échanger avec Craig Parker comporte la séquence de segments facultatifs suivante:

BGM, RFF, NAD, CTA, NAD, CTA et LIN.

De plus, nous avons convenu d'attribuer aux segments un niveau hiérarchique égal à zéro. Par conséquent, aucune répétition de segments, et a fortiori de groupes de segments, n'est admise.

Afin de former un message, les segments doivent être délimités par les segments de début et de fin de message imposés par EDIFACT, à savoir respectivement UNH et UNT (obligatoires). Le diagramme d'un tel type de message est représenté par la figure 8.1.



Figure 8.1: Diagramme du type de message échangé.

Il ne reste plus alors qu'à transformer ce message en interchange, unité d'échange entre partenaires. Pour ce faire, il faut ajouter les segments de début et de fin d'interchange, identifiés respectivement par les segments UNB et UNZ.

Dans la suite de l'exposé, nous avons attribué à ce type de message une sémantique correspondant à celle d'une facture. C'est ce qui justifiera sa dénomination de type

de message INVOIC (pour facture (invoice) en anglais). Pour éviter tout risque de confusion dans l'esprit du lecteur, il nous paraît important de signaler que ce type de message n'est PAS un message UNSM (cf. chapitre 3). Il ne correspond donc pas au type de message INVOIC défini dans le répertoire EDMD (UN/EDIFACT Message Directory). Notons que ce répertoire est en fait absent des fichiers de définition de la norme EDIFACT au sein d'ATLAS. Tout type de message EDIFACT (qu'il soit conforme ou non à l'EDMD) doit donc être défini explicitement par l'utilisateur pour pouvoir être traduit par ATLAS.

8.2. Configuration du logiciel ATLAS-EDI

Dans ce qui suit, nous présentons les étapes à suivre pour configurer le logiciel de façon à traduire des messages du type décrit ci-dessus. Nous formulons l'hypothèse que c'est la première fois que nous accomplissons complètement la démarche.

Nous avons vu précédemment que le logiciel comporte trois grandes parties:

- l'administration du système (System Administration),
- le traitement des données (Data Processing),
- la gestion des communications (Communications).

La partie la plus importante de la configuration concerne l'administration du système. La figure 8.2 montre les différents items du menu déroulant associé au menu *System Administration* de la barre de menus du logiciel. Dans cette partie, nous nous intéresserons seulement aux items *Translations Settings* (validation de certains paramètres pendant les traductions, définition du format des fichiers plats), *Message Table Maintenance* (création et mise à jour des tables de mapping), et *Profile Maintenance* (définition du profil des partenaires).

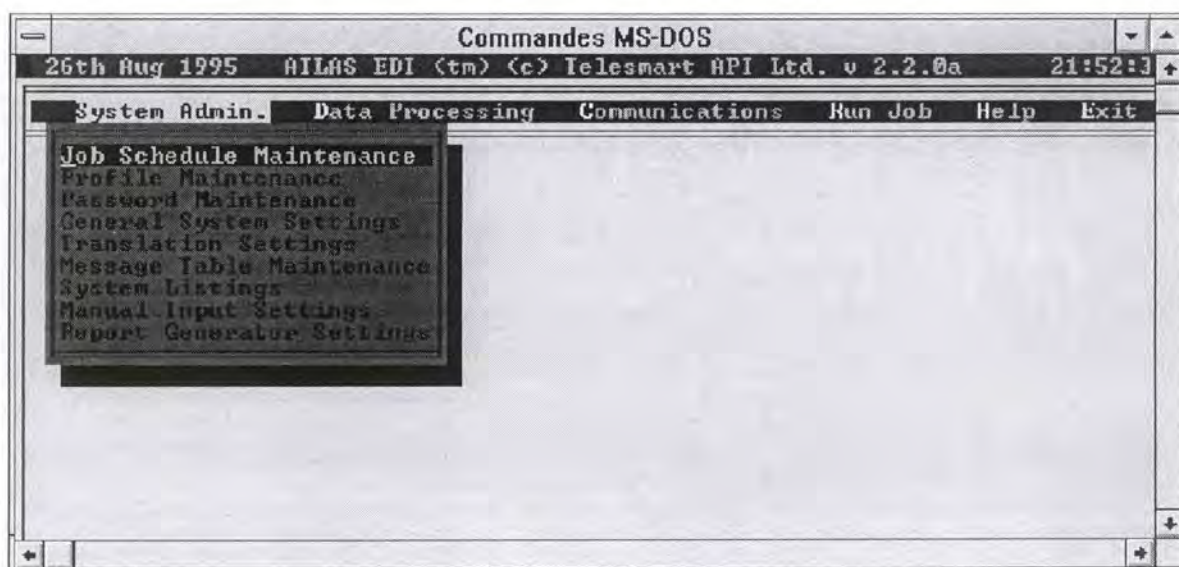


Figure 8.2: Les différents items du menu *System Administration*

La méthode que nous proposons ici est celle qui nous semble la plus logique. Il faut noter que le manuel de l'utilisateur livré avec le logiciel se contente de décrire les différentes fonctionnalités sans guider l'utilisateur dans sa démarche.

Dans un premier temps, nous procédons à la spécification de la syntaxe des fichiers plats de notre application. Cette spécification concerne à la fois les fichiers destinés à être traduits en fichiers EDI (fichiers exportables) et les fichiers issus de la traduction de fichiers EDIFACT (fichiers importés). Conjointement à cette opération, nous devons sélectionner les paramètres que nous souhaitons faire valider automatiquement lorsque les processus de traduction sont enclenchés. Ces deux opérations sont réalisées simultanément parce que le logiciel ne met à la disposition de l'utilisateur qu'un seul écran de saisie de données à cet effet.

Ensuite, il faut spécifier au logiciel la structure du type de message que l'on désire échanger. Sur base de cette spécification, le logiciel construit une table de mapping qu'il faut ensuite compléter afin de satisfaire nos exigences. Cela revient à adapter la table de mapping à la structure des fichiers plats que nous utilisons.

Une fois que la table de mapping est définie, nous complétons le profil de notre partenaire commercial en lui associant la référence du type de message que nous souhaitons échanger avec lui.

A présent, nous allons détailler chacune de ces étapes en établissant les liens qui existent entre elles.

8.2.1. Sélection des paramètres à valider pendant la traduction et spécification de la syntaxe et du format des fichiers plats

La première opération à effectuer lorsque l'on désire configurer le logiciel pour échanger un type de message donné consiste à spécifier la syntaxe et le format des fichiers plats produits et utilisés par l'application. De la sorte, ATLAS-EDI sera capable de les interpréter correctement au moment de la traduction (EXPORT et IMPORT).

Comme nous l'avons mentionné, cette opération est indissociable de la sélection des paramètres que l'on désire faire valider pendant les processus de traduction. En réalité, la sélection des paramètres précède la définition des formats des fichiers plats.

La figure 8.3 montre quelle sélection d'item du menu *Translation Settings* il faut effectuer pour réaliser cette étape. Un sous-menu apparaît dès que l'utilisateur sélectionne l'item *Translation Settings*. Nous nous intéressons ici au premier item de ce sous-menu, *Translation Defaults*.

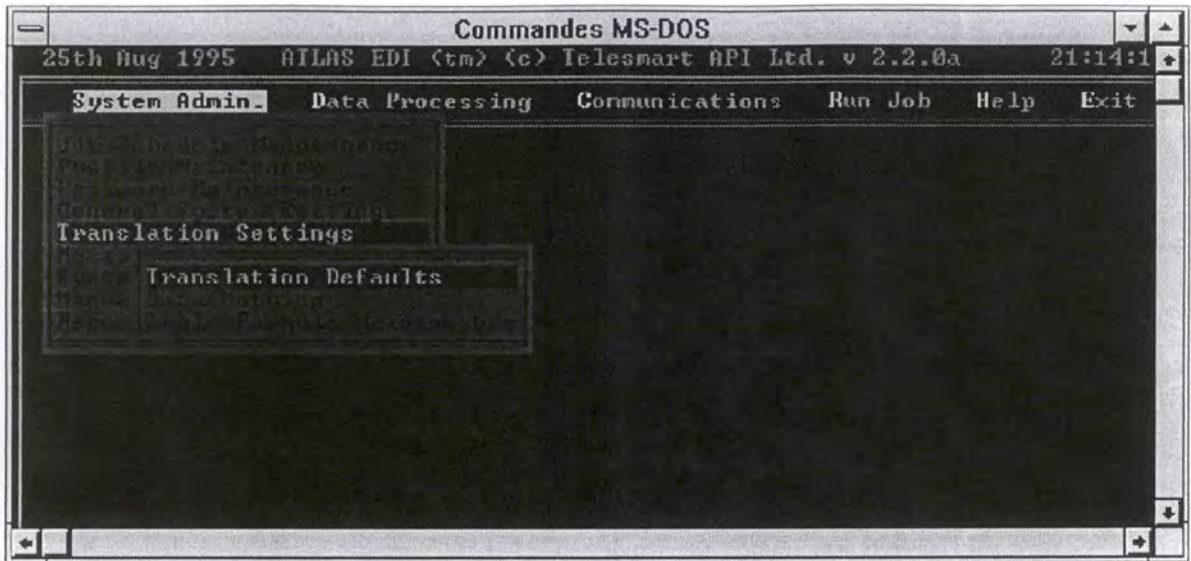


Figure 8.3: Sélection d'item du menu *Translation Settings*.

En sélectionnant l'item *Translation Defaults*, l'écran illustré par la figure 8.4 apparaît. Seuls les menus *Import Details* et *Export Details* nous intéressent ici. En sélectionnant le menu « *Import Details* », l'utilisateur peut définir la structure générale des fichiers plats produits par le logiciel au cours de la traduction IMPORT. Le choix de l'option « *Export Details* » permet à l'utilisateur de définir la structure générale des fichiers plats destinés à être traduits en format EDIFACT.

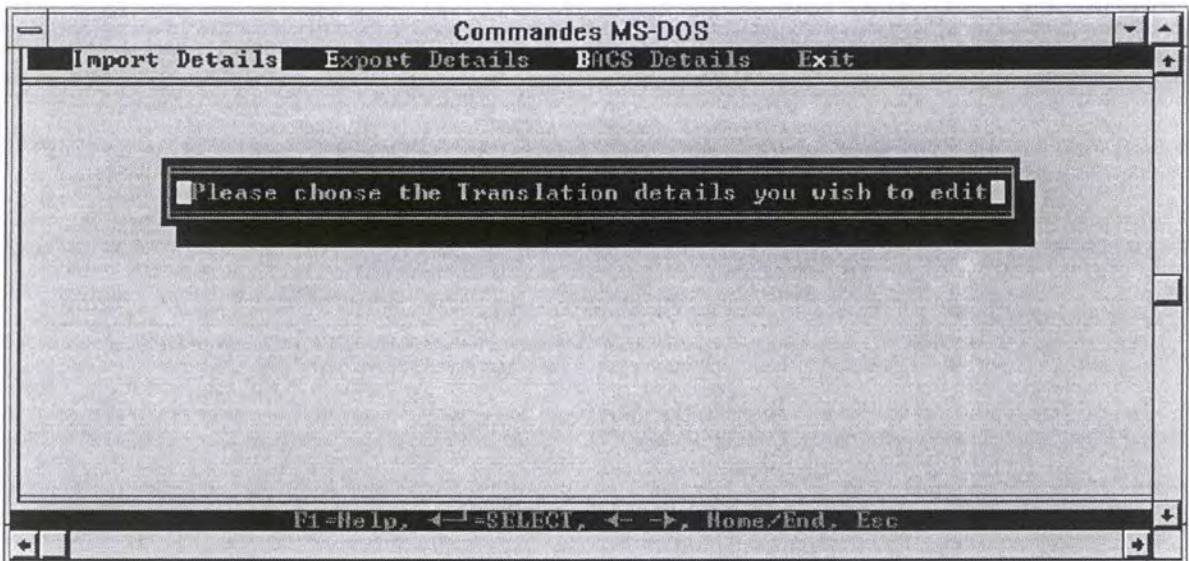


Figure 8.4: Barre de menu de sélection des paramètres de traduction devant être configurés.

8.2.1.1. Sélection des paramètres de la traduction IMPORT et définition de la structure des fichiers plats importés

La figure 8.5 montre l'écran qui apparaît après que l'on ait sélectionné l'item *Import Details*, dans lequel le choix des paramètres à valider (les six premières lignes) a été réalisé. La septième ligne définit le type de structure des fichiers plats et est complétée par un cadre superposé dans lequel l'utilisateur détaille leur structure.



Figure 8.5: Ecran de configuration des paramètres d'IMPORT.

Nous avons choisi les validations suivantes pour la traduction des fichiers EDI en fichiers plats:

- la validation du caractère obligatoire des champs (*Mandatory Field Validation*) est choisie (« Y »). Ceci implique que si des champs obligatoires ne sont pas présents dans le fichier EDI à traduire, le processus de traduction signalera une erreur.
- la validation de la taille des éléments de données du fichier EDI (*Field Size Validation*) est également prescrite. En cas de dépassement de la taille maximum fixée pour chaque élément de donnée par le standard EDIFACT, un message d'erreur est généré.
- la validation des types de champ (*Field Type Validation*) a comme conséquence le signalement d'une erreur lorsqu'un élément de donnée appartenant au fichier EDI est d'un type différent de celui fixé par EDIFACT.
- la validation des groupes de répétitions (*Group Repeat Validation*) permet un contrôle du respect du nombre maximal de répétitions d'un segment dans le fichier EDI. En cas de dépassement de cette limite, une erreur est signalée.
- la validation des listes de codes standards (*Standard List Validation*) est également prescrite. Par conséquent, lors de la traduction d'un fichier EDIFACT, les codes mentionnés dans ce message sont soumis à un contrôle de validité. Si un code n'est

pas conforme à la liste de codes définie dans la version d'EDIFACT employée, une erreur est signalée au cours du processus de traduction.

Un autre paramètre, le *Data Tracking*, n'influence pas la traduction. Il permet, lorsqu'il est validé, de connaître (au moyen de l'item *Data Tracking* sélectionné à partir de l'item *Data Processing* de la barre de menu du logiciel) l'état courant (archivé ou non, ...) d'un message (ou d'un interchange) particulier au sein du système.

La septième ligne définit un des deux types de formats que l'on peut attribuer aux fichiers plats importés:

- soit un format de fichier dont la taille des records est variable,
- soit un format de fichier dont les records sont de taille fixe.

Rappelons que les fichiers plats sont constitués de records, eux-mêmes composés de champs de donnée.

Dans les fichiers plats dont la taille des records est fixe, chaque champ de donnée est de taille fixe. Il est identifié par les positions du premier et du dernier caractère qui lui sont réservés au sein de ce record.

Dans les fichiers plats dont la taille des records est variable, la place occupée par un champ de donnée est le plus souvent variable. Dans ce type de fichier, la fin de chaque champ de donnée est signalée par la présence d'un délimiteur (un caractère particulier normalement absent des éléments de données, un « £ » ou un « ^ » par exemple), et c'est le numéro d'ordre d'un champ de donnée dans la suite des champs de donnée d'un record qui permet de l'identifier. Ce système impose bien entendu la présence d'un délimiteur même lorsque un champ de donnée ne comprend aucun élément de donnée. (Bien qu'il ne s'agisse pas de fichiers plats applicatifs, les fichiers des messages EDI du standard EDIFACT sont typiquement des fichiers à records de taille variable).

L'avantage d'un fichier plat à taille de records fixe est qu'il permet une localisation plus rapide de chaque élément de donnée. Si un processus de traduction doit accéder à un champ de donnée particulier, il pourra y accéder de manière directe en se basant sur sa position physique au sein du record. Par contre, dans le cas d'un fichier à taille de records variable, il devra parcourir le record du début jusqu'au champ de donnée recherché. Si celui-ci est le dernier du record et que le nombre de records à traiter est important, la perte de temps peut être appréciable.

L'avantage d'un fichier à records de taille variable, c'est qu'il permet une réduction drastique de la taille des records lorsque ceux-ci comportent de nombreux champs de donnée facultatifs. Prenons l'exemple d'un segment NAD que nous voudrions représenter dans un fichier à records de taille fixe. La figure suivante nous donne une liste exhaustive des éléments de donnée pouvant appartenir à ce segment.

Nom de l'élément de donnée	Code	Type	Taille (maxi)
		M:obligatoire C:facultatif	
NAD001 NAME AND ADDRESS	NAD	C	
PARTY QUALIFIER	NAD3035	M	3
NAME AND ADDRESS	NADC082	C	
PARTY IDENTIFICATION, CODED	NAD3039 1	M	17
CODE LIST IDENTIFIER, CODED	NAD1131 2	C	2
NAME AND ADDRESS	NADC058	C	
NAME AND ADDRESS LINE	NAD3124 1	M	35
NAME AND ADDRESS LINE	NAD3124 2	C	35
NAME AND ADDRESS LINE	NAD3124 3	C	35
NAME AND ADDRESS LINE	NAD3124 4	C	35
NAME AND ADDRESS LINE	NAD3124 5	C	35
NAME AND ADDRESS	NADC080	C	
PARTY NAME	NAD3036 1	M	35
PARTY NAME	NAD3036 2	C	35
PARTY NAME	NAD3036 3	C	35
NAME AND ADDRESS	NADC059	C	
STREET AND NUMBER/ P.O. BOX	NAD3042 1	M	35
STREET AND NUMBER/ P.O. BOX	NAD3042 2	C	35
STREET AND NUMBER/ P.O. BOX	NAD3042 3	C	35
CITY NAME	NAD3164	C	35
COUNTRY SUB-ENTITY, CODED	NAD3229	C	9
POST CODE	NAD3251	C	9
COUNTRY, CODED	NAD3207	C	2
$\Sigma = 462$			

Si nous voulions représenter un segment NAD par traduction de celui-ci en un record de fichier plat de longueur fixe capable de représenter l'ensemble de ses éléments de données, celui-ci devrait avoir une longueur minimale de 462 caractères !

Par contre, dans un fichier plat à records de taille variable capable de représenter l'ensemble des éléments de données du segment NAD, la taille du record sera généralement bien inférieure. En effet, une donnée facultative absente n'entraînera

l'apparition que d'un seul caractère au sein du fichier plat, celui correspondant au délimiteur qui lui est associé.

Notons que, dans les deux cas, la taille du fichier plat peut toujours être réduite si l'on décide de ne pouvoir y insérer qu'une partie des éléments de données facultatifs définis par le standard (les éléments de données obligatoires devront bien entendu toujours en faire partie).

Afin d'illustrer notre propos, voici la structure de deux fichiers plats, un dont les records sont de taille fixe, l'autre dont ceux-ci sont de taille variable.

Le premier est un fichier plat dont les records sont de taille fixe:

TG1		ORG_ID		INTROR	XX
UNH	DEM5	INTROR			
REF	UCN	DEM5			
REF	CDN	CUSREF			
NAD	CN	ORG_ID	ZZ	AIKASHINKOHOLDINGSINC.	
NADA		1-11-3	SHIBA-DAIMON	TOKONAKA CITY	
NADB				JAPAN	
NAD	EX	ORG_ID		ESPANOLA DE PINTORA S.A.	
NADA		AVE SALAMANCA 13		OLIVENZIA, BILBAO, SPAIN	
REF2	EXN	SPEX/DEM/888B/22			
REF2	VAT	VAT1			
PROD		FRESH MILK CARTONS		CARTON	5200
CODE		CAP04	001		
PROD		EGGS - SALMONELLA FREE		CASE	4500
CODE		CAP01	002		

On remarque l'absence de délimiteur ainsi que la présence des éléments de données à des positions bien définies.

Ce deuxième fichier est un fichier plat dont les records sont de taille variable (c'est le type de fichier plat que nous allons utiliser lors de nos traductions):

```

ENV001^INVOICES^REF02^^12345^^1111^^^^^^^^^^^^^^1
INVO01^1^INVOIC^1^901^UN^^^^
BGM001^380^INVOIC^INV001^950711^^00^^^^
RFF001^PO^PO0012^^950302^
NAD001^SE^STUDENTS^^NAMUR UNIVERSITY^^^^^^^^^^^^^^BE
CTA001^AD^^ACCOUNTING DEPARTMENT^^^^^^^^^^^^
NAD002^BY^^^^^^^^^^^^^^^^
CTA002^PD^^PURCHASING PERSON^^^^^^^^^^^^
LIN001^^0200^VN^^01^10^TV^2000^^^^^^^^^^^^
UNT001^9^2
UNZ001^1^REF02

```


On remarque la présence d'un délimiteur (un caractère « ^ ») entre chaque élément de donnée. La présence de délimiteurs accolés signale l'absence (au sein de cette instance du fichier plat) d'éléments de données pouvant être intégrés dans ce type de fichier plat.

Le fait de choisir une taille variable pour les records provoque l'apparition du cadre en superposition représenté à la figure 8.5. Dans ce cadre, nous avons choisi le caractère « ^ » pour séparer les champs successifs d'un même record. De plus, ces records sont séparés par un retour à la ligne (symbolisé par un « D »). Enfin, il n'existe aucun caractère spécial entre le dernier champ d'un record et le caractère qui marque la fin du record (fin de ligne).

En effectuant cette série de choix, nous venons de définir les seuls éléments du format général des fichiers plats qui puissent être spécifiés par l'utilisateur. La correspondance entre les records du fichier plat et ceux du fichier EDIFACT correspondant est triviale et ne peut être modifiée par l'utilisateur: à un record du fichier EDI correspond un et un seul record dans le fichier plat. De plus, l'ordonnancement des records est le même dans les deux fichiers. La correspondance entre les éléments de données d'un record du fichier plat et ceux du record correspondant sera quant à elle définie dans les tables de mapping.

Le fichier plat reproduit ci-dessus satisfait bien aux caractéristiques générales que nous venons de définir.

Notons que ces spécifications sont générales pour l'ensemble du système, elles s'appliquent à tous les fichiers plats importés par le logiciel, quels que soient les partenaires, les standards et les types de messages utilisés.

Si nous choisissons le type de structure de fichiers plats caractérisé par des records de taille fixe, les paramètres à compléter sont les suivants (cf. figure 8.6):

- la longueur des records (celle-ci est fixée par défaut à une valeur de 80),
- la longueur du label du record (ce label identifiant du record se situant toujours en début de record, cette information suffit à le repérer sans ambiguïté).
- le séparateur de record (un caractère de retour à la ligne est proposé par défaut).

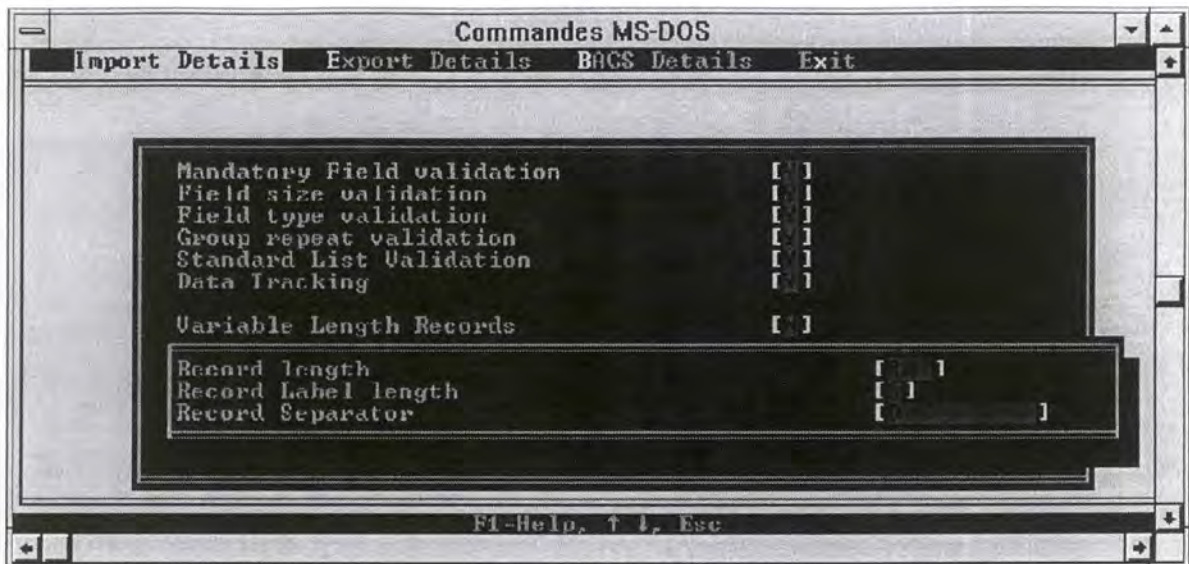


Figure 8.6: Boite de sélection des paramètres syntaxiques des fichiers plats importés à records de taille fixe.

8.2.1.2. Sélection des paramètres de la traduction EXPORT et spécification de la structure des fichiers plats à exporter

Le principe suivi ici ressemble très fort à celui appliqué au cas des fichiers plats importés. La grande différence entre les deux démarches réside dans la nécessité de détailler davantage la structure des fichiers plats à exporter. En effet, ATLAS EDI doit non seulement les identifier mais aussi y trouver les renseignements qui permettent au processus de traduction EXPORT de savoir:

- à quel partenaire (identifié par une référence unique à spécifier dans son profil) est destinée l'information à traduire;
- de quel document échangé avec ce partenaire il s'agit (comme nous le verrons, cette information définit aussi quelle table de mapping utiliser lors de la traduction).

La sélection de l'item *Export Details* représenté à la figure 8.4 provoque l'affichage d'un écran de saisie de données. La figure 8.7 illustre l'état de cet écran après avoir spécifié les paramètres de la traduction EXPORT et le format des fichiers plats exportables.

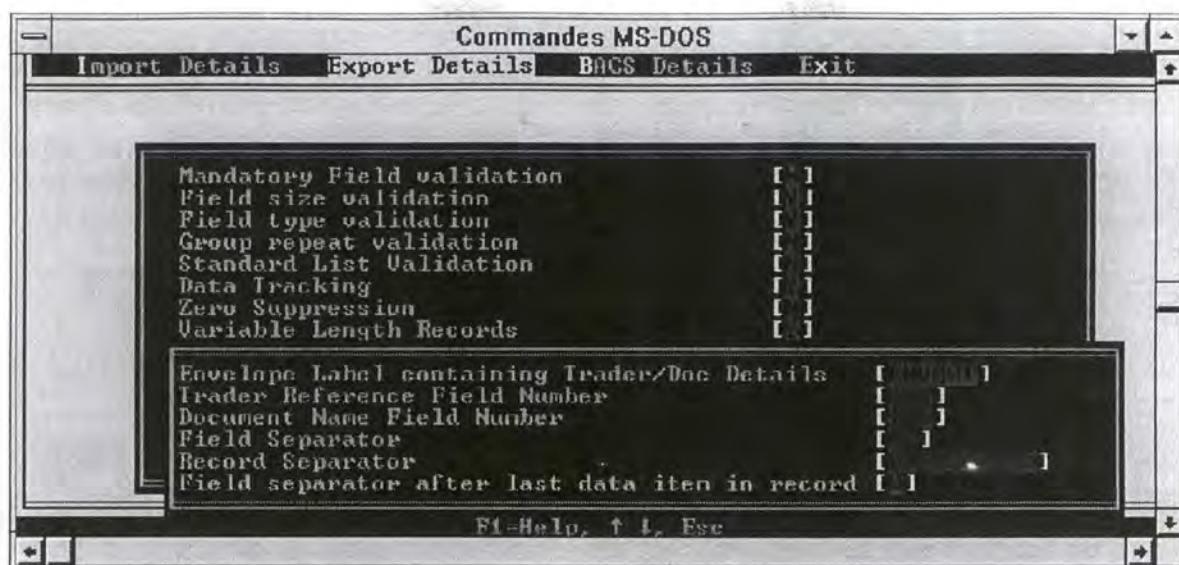


Figure 8.7: Boite de sélection des paramètres syntaxiques des fichiers plats à exporter à records de longueur variable.

Les validations proposées sont identiques à celles vues dans le cas de l'*Import Details*, à l'exception de l'option « Zero Suppression » qui permet d'enlever les zéros non significatifs des données numériques avant de les recopier dans le fichier EDIFACT.

Nous avons choisi une structure de type « Variable Length Records » pour les fichiers plats exportables, comme nous l'avons fait pour les fichiers plats importés. Notons qu'il n'existe a priori aucune obligation de définir le même type de records (de taille variable ou de taille fixe) pour les fichiers plats importés et exportables: les fichiers plats importés peuvent être d'un format différent de celui des fichiers destinés à être émis par notre application.

Comme nous l'avons déjà mentionné, les fichiers plats exportables doivent être identifiés de façon précise par le logiciel. En effet, pour effectuer la traduction EXPORT d'un fichier plat, le logiciel a besoin de connaître la référence du partenaire destinataire de l'information à traduire et le nom du document échangé avec ce partenaire. Ces deux informations, nécessaires pour que le logiciel puisse choisir une table de mapping, sont contenues dans le premier record du fichier plat à traduire. Ce premier record correspond en fait au premier segment de l'interchange EDIFACT. Il faut tout d'abord mentionner le nom qui identifie le record contenant ces informations. Il reste ensuite à renseigner la position dans ce record (position relative au record label) des champs contenant la référence du partenaire et le nom du document. Comme nous le verrons, dans le profil du partenaire, l'identifiant de la table de mapping à utiliser est associé au nom du document échangé.

Voici, par exemple, un fichier plat à exporter qui répond à la définition qui vient d'être établie:

```
ENV001^INVOICES^^UNOA^1^12345^^EDIUSER^^940815^1656^2^~~~~~1
INVO01^1^INVOIC^1^901^UN~~~~
BGM001^380^INVOIC^INV001^950711^^00~~~~
RFF001^PO^PO0012^^950302^
NAD001^SE^STUDENTS^^NAMUR UNIVERSITY~~~~~BE
CTA001^AD^^ACCOUNTING DEPARTMENT~~~~~
NAD002^BY~~~~~
CTA002^PD^^PURCHASING PERSON~~~~~
LIN001^^0200^VN^^01^10^TV^2000~~~~~
UNT001^9^1
UNZ001^1^2
```

Dans le cas illustré ici, pour tout fichier plat destiné à être traduit en fichier EDIFACT, la référence du partenaire (en l’occurrence « 12345 ») se trouve dans le cinquième champ du record identifié par le label « ENV001 », et le nom du document à traduire se trouve dans le champ qui suit immédiatement cette étiquette (position 1).

A titre de comparaison, la figure 8.8 illustre l’écran qui permet de définir des fichiers à exporter dont la taille des records est fixe.



Figure 8.8: Boite de sélection des paramètres de format général des fichiers plats à exporter à records de longueur fixe.

De manière analogue au cas des fichiers à records de taille variable, nous devons commencer par spécifier le label du record contenant les références du partenaire et du document à traduire. Nous signalons également la position et la taille de ces références. Pour terminer, nous entrons la taille des records, la longueur de leur label ainsi leur caractère séparateur.

Une fois que la définition de la structure des fichiers plats est terminée, il faut construire puis éditer une table de mapping qui sera utilisée lors de la traduction

(IMPORT et EXPORT) de fichiers EDIFACT et de fichiers plats internes à l'application.

8.2.2. Construction d'une table de mapping associée au type de message étudié

Une table de mapping est toujours associée à un type de message et renseigne notamment:

- la liste complète des segments et de leurs éléments de données,
- le caractère obligatoire ou facultatif des segments et de leurs éléments de données,
- pour chaque segment, le record label du record correspondant (dans les fichiers plats importés et à exporter),
- la position dans les fichiers plats importés de la valeur des différents éléments de données extraits d'un message EDI traduit,
- la position dans les fichiers plats à exporter de la valeur des éléments de données à recopier dans les fichiers EDI.

En vertu de ce qui vient d'être dit, une table de mapping associée à un type de message doit être construite notamment sur base de la composition en segments de ce type de message. La séquence des segments ne suffit pas à définir seule un type de message, il faut aussi définir leur statut (niveau hiérarchique, caractère obligatoire ou facultatif, existence éventuelle de répétitions). La définition complète du type de message précède la construction de la table de mapping associée et l'édition de celle-ci.

La définition du type de message qui nous intéresse est réalisée en sélectionnant l'item *Message Table Maintenance* indiqué à la figure 8.9 puis l'item *Message Table Generator* du sous-menu correspondant.



Figure 8.9: Sélection de l'item de définition du type de message.

La définition d'un type de message s'effectue en deux temps. Dans le premier, il faut spécifier les caractéristiques du message (nom, standard, syntaxe, ...) comme l'indique la figure 8.10. Ensuite, il faudra donner sa structure, définie par la séquence des segments qu'il contient.

Dans l'écran illustré par la figure 8.10, nous avons nommé le type de document « INVOICES » et le type de message « INVOIC ».

Table Header Information	
Document Type	[INVOICES]
Message Type	[INVOIC]
Document Description	[INVOICE MESSAGE]
Standard	[EDIFACT]
Syntax	[X12]
Element Directory	[UNEDIFACT]
Sequence	[1]
No. of messages in document	[1]
Version Identifiers	[1]
Filename	[INVOICES]

F1-Help, F2-Edit, ↑↓, Esc

Figure 8.10: Ecran de saisie des caractéristiques générales du type de message à définir.

Rappelons que la notion de document est identique à la notion de message dans le cas du standard EDIFACT. Dans d'autres standards (comme par exemple le standard TRADACOMS), un document est éventuellement défini par plusieurs messages. Il s'agit alors de spécifier plusieurs messages qui se rapportent au même document. Comme les écrans de saisie des informations sont génériques, on est obligé de donner un nom de document, même si l'on sait que ce document se confond au message. La description du document que nous renseignons est « invoice message », le standard et la syntaxe utilisés sont EDIFACT. Le répertoire des éléments est celui correspondant à la version 90.1 d'EDIFACT. Le numéro de séquence (équivalent au numéro d'ordre du message dans la série de messages pouvant constituer le document) est « 1 » (puisque le document ne comporte qu'un message), le nombre de messages contenus dans un document est « 1 » et l'identifiant complet de la version d'EDIFACT est formé des éléments suivants (cf. fig 8.11):

- le numéro de version est égal à 1. Cela signifie qu'aucun changement n'a été apporté à la spécification originale d'EDIFACT. Si l'on changeait par exemple le type d'un des éléments de données, le numéro de version serait différent.
- le numéro de Release est 901 (premier semestre de l'année 1990).
- le nom de l'institution qui contrôle le standard (United Nations).

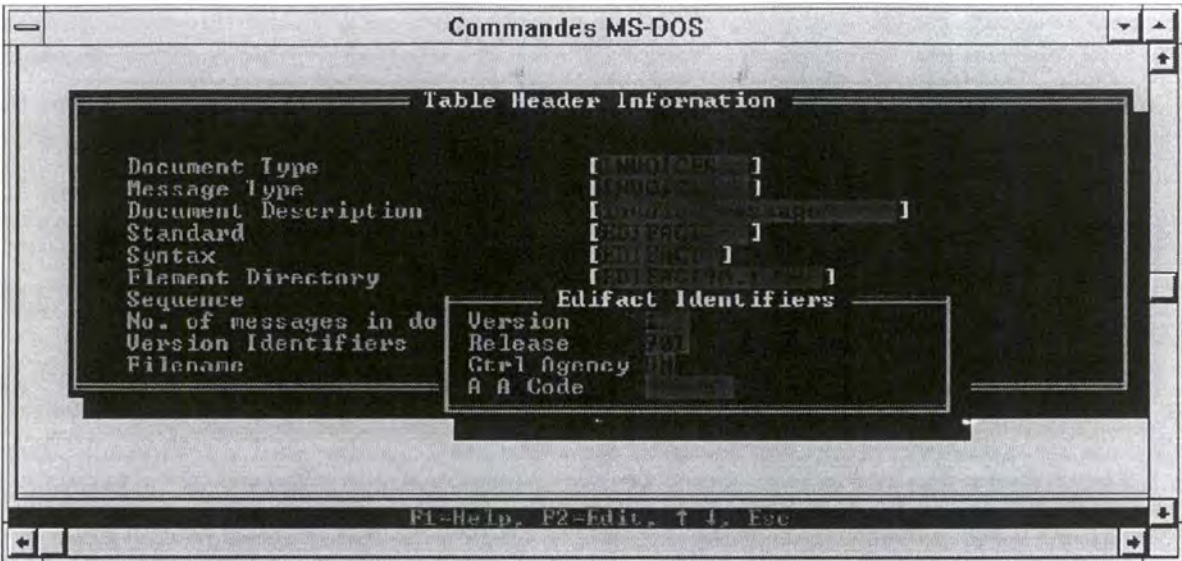


Figure 8.11: Boite d'identification de la version d'EDIFACT utilisée.

Ces identificateurs sont représentés sous la forme condensée « 1 901UN » à la figure 8.10.

Enfin, nous donnons le nom « INVOICE » au fichier qui contiendra la définition de ce type de message (il est automatiquement nommé INVOICE.TBL). L'écran représenté à la figure 8.12 est destiné à construire la séquence des segments constitutifs du type de message. Nous sélectionnons chaque segment dans une liste présentée en superposition (elle contient 51 segments). Cette liste de segments est bien sûr conforme à la version d'EDIFACT précisée dans l'écran précédent.

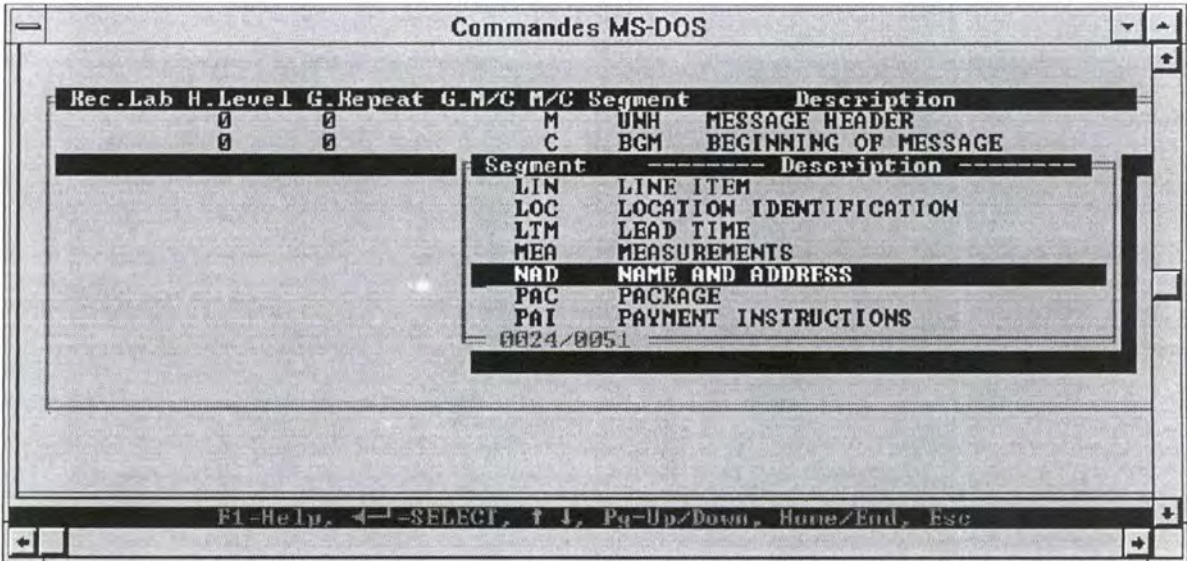


Figure 8.12: Ecran de saisie de la séquence des segments constitutifs du type de message.

Chaque fois qu'un segment est sélectionné, les données suivantes s'affichent dans le cadre TABLE GENERATOR:

- le niveau hiérarchique du segment (Hierarchical Level), mis à « 0 » par défaut;
- le nombre maximum de fois que ce segment peut être répété dans un groupe de segments (selon leur niveau hiérarchique). La valeur qui s'affiche par défaut est « 0 », ce qui est normal puisque le niveau hiérarchique est lui aussi mis à « 0 ».
- le caractère obligatoire ou optionnel de la présence de ce segment dans le type de message (M/C). Par défaut, tout segment est déclaré optionnel (« C »).
- l'étiquette correspondant à ce segment (Segment);
- la description du segment.

Ces valeurs sont automatiquement affichées après chaque sélection d'un segment. L'utilisateur est libre de les éditer. Par exemple, il peut changer les niveaux hiérarchiques et définir un nombre maximum de répétitions d'un segment au sein d'un groupe. Ici, nous avons défini les segments UNH et UNT comme étant obligatoires (« M »). La figure 8.13 montre l'écran obtenu après avoir défini complètement la structure du type de message.

Rec.Lab	H.Level	G.Repeat	G.M/C	M/C	Segment	Description
	0	0		M	UNH	MESSAGE HEADER
	0	0		C	BGM	BEGINNING OF MESSAGE
	0	0		C	RFF	REFERENCES
	0	0		C	NAD	NAME AND ADDRESS
	0	0		C	CTA	CONTACT SEGMENT
	0	0		C	NAD	NAME AND ADDRESS
	0	0		C	CTA	CONTACT SEGMENT
	0	0		C	LIN	LINE ITEM
	0	0		M	UNT	MESSAGE TRAILER

Figure 8.13: Liste des segments sélectionnés pour le type de message.

Il reste deux types d'informations (non précisées par défaut) à donner à propos des segments:

- le Record Label (premier en-tête du cadre du TABLE GENERATOR) correspondant aux segments n'est pas indiqué par défaut et n'est pas éditabile ici (il faut attendre l'étape d'édition de la table de mapping). Ces Record Labels apparaissent dans le fichier plat obtenu après la traduction du fichier EDIFACT et servent à référencer les segments correspondants de ce fichier EDIFACT.
- le caractère obligatoire (Mandatory) ou optionnel (Conditional) de la répétition du segment dans un groupe (Group M/C). Cette information est à éditer manuellement pour chaque segment lorsque les valeurs de son niveau hiérarchique et du nombre maximum de répétitions dans un groupe sont toutes les deux supérieures à zéro. Ici,

comme aucune répétition n'est admise, il est insensé (et même impossible) d'éditer les valeurs correspondant à ce paramètre.

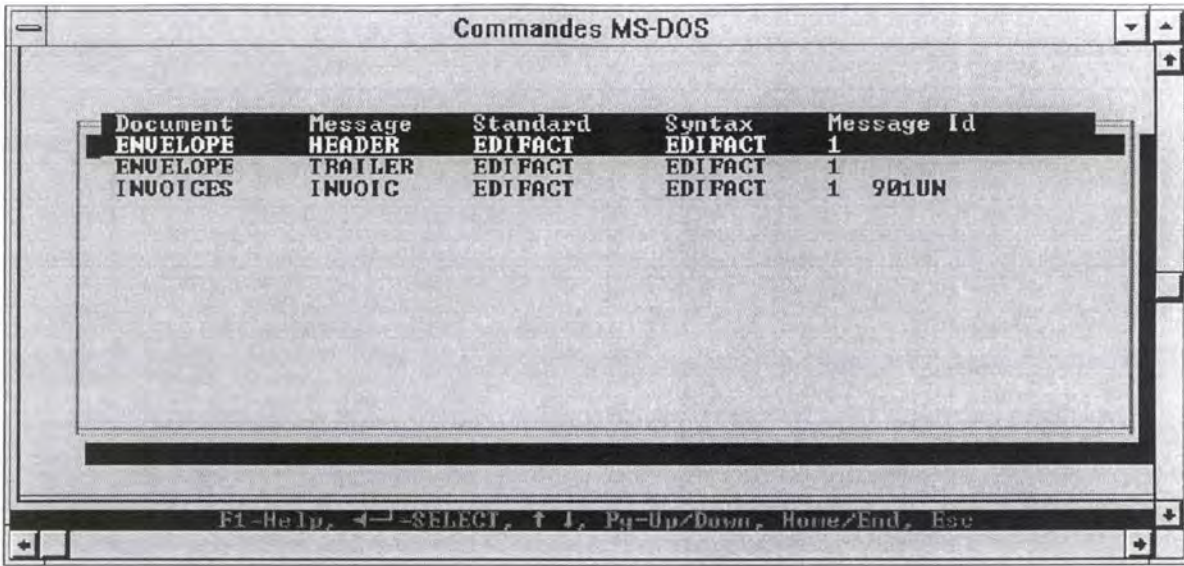
Au moment où l'utilisateur demande d'enregistrer les informations qu'il a rentrées, le logiciel affiche un message qui lui indique qu'il va créer une table de mapping pour le type de message créé. Ainsi, il ne reste plus qu'à éditer cette table au cours de l'étape suivante.

8.2.3. Edition de la table de mapping associée au type de message

A ce stade, le logiciel dispose de la définition du type de message (fichier INVOICE.TBL) auquel nous devons associer une table de mapping à définir. Nous avons vu qu'une table est créée automatiquement au moment de l'enregistrement de la définition du type de message. Il faut à présent éditer cette table afin de l'adapter à nos besoins. En effet, il faut enregistrer les informations suivantes:

- pour chaque segment, le record label du record correspondant (dans les fichiers plats importés et à exporter),
- la position dans les fichiers plats importés de la valeur des différents éléments de données extraits des messages EDI traduits,
- la position dans les fichiers plats à exporter de la valeur des éléments de données à recopier dans les fichiers EDI.

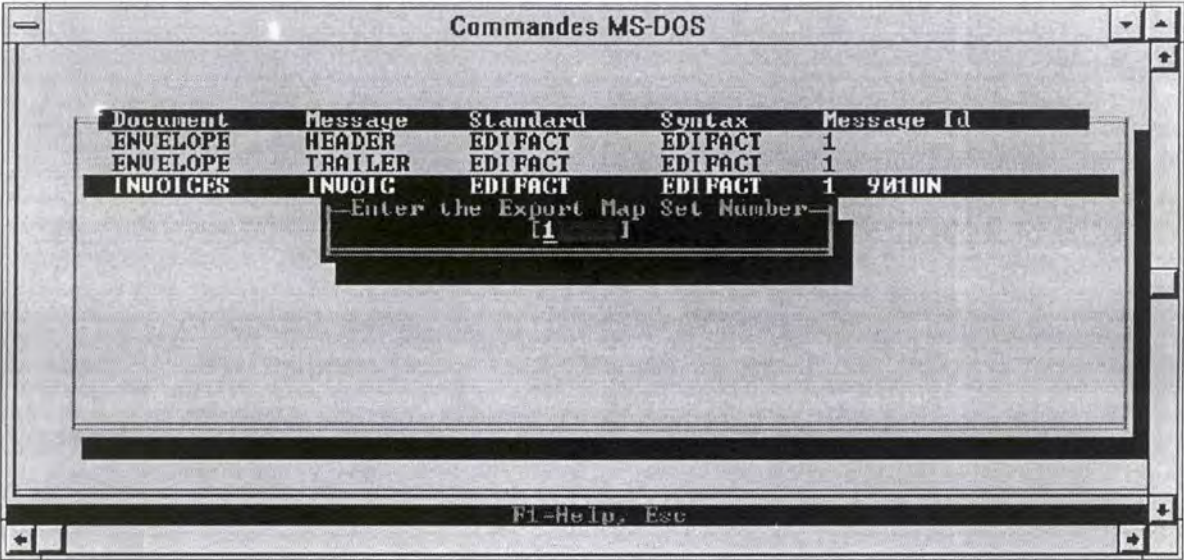
Il faut retourner au menu déroulant du *System Administration*, et y sélectionner l'item *Translation Settings* puis l'item *Data Mapping* de son sous-menu. La liste des documents (messages) pour lesquels une table de mapping existe apparaît à l'écran illustré par la figure 8.14. On sélectionne le message INVOIC dont l'identifiant correspond à celui (« 1 901UN ») défini lors de l'étape précédente (TABLE GENERATOR). A ce moment, un cadre apparaît superposé à l'autre, illustré à la figure 8.15. Le système nous demande d'introduire le numéro (Export Map Set Number) qui identifie la version de la table de mapping associée au message INVOIC. En effet, ATLAS nous permet de définir plusieurs versions de table de mapping pour un même message.



The screenshot shows a window titled "Commandes MS-DOS" with a table of message types. The table has five columns: Document, Message, Standard, Syntax, and Message Id. The rows are: ENVELOPE, ENVELOPE, INVOICES, HEADER, TRAILER, and INVOIC. The Message Id for INVOICES is 901UN. The bottom status bar shows keyboard shortcuts: F1-Help, ←-SELECT, ↑↓, Pg-Up/Down, Home/End, and Esc.

Document	Message	Standard	Syntax	Message Id
ENVELOPE	HEADER	EDIFACT	EDIFACT	1
ENVELOPE	TRAILER	EDIFACT	EDIFACT	1
INVOICES	INVOIC	EDIFACT	EDIFACT	1 901UN

Figure 8.14: Liste des types de messages définis dans le système.



The screenshot shows the same window as Figure 8.14, but with an additional dialog box. The dialog box has the title "Enter the Export Map Set Number" and a text input field containing the number "1". The bottom status bar shows keyboard shortcuts: F1-Help and Esc.

Document	Message	Standard	Syntax	Message Id
ENVELOPE	HEADER	EDIFACT	EDIFACT	1
ENVELOPE	TRAILER	EDIFACT	EDIFACT	1
INVOICES	INVOIC	EDIFACT	EDIFACT	1 901UN

Figure 8.15: Sélection d’une table de mapping associée à un type de message.

En réalité, seule la partie export de la table de mapping peut varier entre les différentes tables de mapping permettant de traduire un même type de message. L'utilité des Export Map Set peut être justifiée de la façon suivante: il est fréquent d'échanger le même type de message avec différents partenaires. De son côté, un partenaire peut avoir besoin de certaines informations particulières dont d'autres n'auront aucun usage. Idéalement, il faudrait donc définir une nouvelle table de mapping pour chaque partenaire, ce qui nous permettrait d'extraire des données différentes du fichier plat en fonction du partenaire. L'existence de ces différentes versions d'Export Map Set évite ce travail fastidieux. Habituellement, seul un Export Map set est créé par message. Dans ce cas, on a affaire à un Export Map Set égal à 1, qui est la valeur par défaut lorsque l'on sélectionne des documents. Lors d'une traduction, le choix de l'Export Map Set à utiliser se fera bien évidemment

automatiquement sur base de l'Export Map Set mentionné pour ce type de message dans le profil du partenaire commercial.

La figure 8.16 illustre l'écran qui contient le début de la table de mapping pour le message INVOIC, après avoir procédé à sa définition. Les en-têtes des colonnes sont les suivantes:

- le Record Label: c'est ici que l'on peut éditer les record labels servant à référencer dans les fichiers plats les segments EDIFACT traduits. Ainsi, on peut savoir à quels segments EDIFACT se rapporte l'information que l'on trouve dans les records des fichiers plats.
- la description de chacun des segments et de leurs éléments de données;
- le code (Segment Element) de chaque segment et de leurs éléments de données;
- pour chaque élément de donnée composite, le numéro d'ordre de ses constituants (Su No);
- le type des éléments de données;
- la taille (maximum) de chaque élément de donnée;
- le « numéro d'import » (Import Number) assigné à chaque élément de donnée: ce numéro correspond à la position dans le record (identifié par le record label) où cet élément de donnée sera placé dans le fichier plat au cours de la traduction IMPORT. Par conséquent, en lisant ces numéros, on sait dire exactement quelles informations se trouvent à quels endroits dans le fichier plat obtenu après traduction.
- le « numéro d'export » (Export Number) assigné à chaque élément de donnée: ce numéro correspond à la position dans le record (identifié par le record label) d'où cet élément de donnée est extrait du fichier plat au cours de la traduction EXPORT.

Record Label	Description	Segment Element	Su No	Typ	Siz	Map Set Imp No	Exp No
INVO01	MESSAGE HEADER	UNH		M			
	MESSAGE REFERENCE NUMBER	UNH0062		MUX	14	1	1
	MESSAGE HEADER	UNH009		M			
	MESSAGE TYPE	UNH0065	1	MUX	6	2	2
	MESSAGE VERSION NUMBER	UNH0052	2	MUX	3	3	3
	MESSAGE RELEASE NUMBER	UNH0054	3	MUX	3	4	4
	CONTROLLING AGENCY	UNH0051	4	MUX	2	5	5
	ASSOCIATION ASSIGNED CODE	UNH0057	5	CUX	6	6	6
	COMMON ACCESS REFERENCE	UNH0068		CUX	35	7	7
	MESSAGE HEADER	UNH010		C			
	SEQUENCE OF TRANSFERS	UNH0070	1	MUN	2	8	8
	FIRST AND LAST TRANSFER	UNH0073	2	CFA	1	9	9

Figure 8.16: La table de mapping pour le message INVOIC.

Nous avons édité manuellement les record labels, et nous voyons que le record label qui correspond au premier segment UNH du message est « INVO01 ». Ce label est destiné à identifier dans le fichier plat le record correspondant au début du message. Le segment UNH est défini par une série d'éléments de données, simples et

composites, décrits dans cette figure. Remarquons, par exemple, que l'élément de donnée « Message Header » est un élément composite constitué de cinq sous-éléments numérotés de 1 à 5 (colonne Su No). Le type des éléments de données est également indiqué, selon la convention suivante:

Type	Code
Alphabétique	A
Numérique	N
Date	D
Heure	T
alphanumérique	X
Taille variable	V
Taille fixe	F
Facultatif	C
Obligatoire	M

Par souci de simplicité nous avons attribué les mêmes numéros d'import et d'export pour chaque élément de donnée. Cela signifie que l'on trouve chaque élément de donnée à la même position dans un fichier plat importé que dans un fichier plat exportable. Bien évidemment, la position des éléments de données pourrait très bien différer entre les deux types de fichiers plats. De plus, nous aurions pu décider de ne pas transférer certains éléments de données. Si le numéro d'import d'un élément de donnée est absent, celui-ci ne sera pas extrait de son segment EDIFACT pour être recopié sur le fichier plat. De manière analogue, si le numéro d'export d'un élément de donnée est absent, celui-ci ne sera pas extrait du fichier plat pour être recopié dans le segment EDIFACT correspondant.

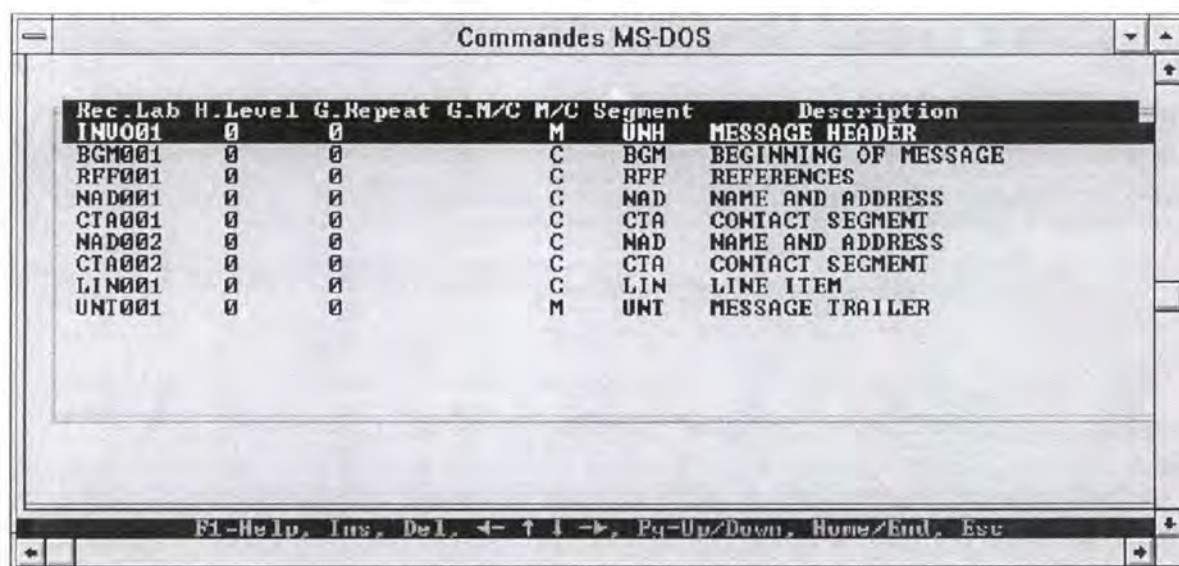
D'autre part, le fait que les numéros d'import et d'export sont attribués dans un ordre croissant implique que l'ordre des éléments de données dans les fichiers plats est le même que celui imposé par EDIFACT. Il se pourrait fort bien que cet ordre ne soit pas respecté et que, par exemple, un élément de donnée de position n dans un segment EDIFACT se retrouve à une position m ($m < n$) dans des fichiers plats. Il s'agit donc ici d'un cas particulier.

Pour rappel, il n'est pas question d'attribuer aux records un ordre qui diffère de celui des segments. Ainsi, la séquence des records d'un fichier importé respecte scrupuleusement la séquence des segments du message EDIFACT traduit. Cette propriété, propre aux tables de mapping d'ATLAS (cf. chapitre 10), est la même pour les fichiers plats exportables: l'ordre de leur records doit être rigoureusement identique à celui des segments correspondants dans le message EDIFACT.

Dans le coin supérieur droit se trouve la valeur de l'Export Map Set qui identifie la table de mapping. Si l'on désire définir une autre table de mapping pour le message INVOIC, il suffit de l'enregistrer avec un numéro d'Export Map Set différent, par exemple égal à 2. Ainsi, il est possible de définir un nombre a priori illimité de tables de mapping pour guider la traduction en export d'un seul type de message. Ces tables pourront donc différer par les valeurs de leurs numéros d'export, leurs numéros d'import et leurs record labels restant identiques. De plus, ces différentes versions de tables de mapping doivent toujours se conformer aux spécifications du type de message contenues dans le fichier INVOICE.TBL. Il ne s'agit donc pas de changer le type et l'ordre des segments par exemple. De tels changements nécessitent la définition d'un nouveau type de message.

Nous verrons plus loin comment affecter une seule version d'une table de mapping pour la traduction d'un type de message avec un partenaire.

Nous avons dit que c'est au cours de cette étape de définition de la table de mapping que l'on peut éditer les record labels. En fait, dès que l'on enregistre les modifications de la table de mapping, les labels édités sont automatiquement recopiés dans le cadre MESSAGE GENERATOR en dessous de l'en-tête Record Label. La figure 8.17 illustre cette mise à jour où on constate que les labels correspondent à ceux indiqués dans la table de mapping.



Rec.Lab	H.Level	G.Repeat	G.M/C	M/C	Segment	Description
INVO01	0	0		M	UNH	MESSAGE HEADER
BGM001	0	0		C	BGM	BEGINNING OF MESSAGE
RFP001	0	0		C	RFP	REFERENCES
NAD001	0	0		C	NAD	NAME AND ADDRESS
CTA001	0	0		C	CTA	CONTACT SEGMENT
NAD002	0	0		C	NAD	NAME AND ADDRESS
CTA002	0	0		C	CTA	CONTACT SEGMENT
LIN001	0	0		C	LIN	LINE ITEM
UNT001	0	0		M	UNT	MESSAGE TRAILER

Figure 8.17: Mise à jour des record labels des segments du type de message considéré.

Bien que la définition d'une table de mapping pour le type de message qui nous intéresse soit une étape cruciale, il ne faut pas oublier qu'il faut aussi définir une table pour les segments UNB et UNZ de début et de fin d'interchange respectivement. Ces tables sont déjà définies pour les enveloppes EDIFACT au moment où on installe le logiciel. Il ne reste plus qu'à changer éventuellement quelques valeurs dans les tables pour qu'elles soient cohérentes avec le format de nos fichiers plats.

Pour le segment UNB (segment de début d'interchange), il faut que les valeurs assignées aux Export et Import Numbers des éléments de données soient en accord avec les positions mentionnées dans l'étape de définition du format des fichiers plats. Pour rappel, au sein du record identifié par le label « ENV001 » (c'est le record correspondant au segment UNB), le nom du document à traduire se trouve dans le champ numéro 1 et la référence du partenaire se trouve dans le champ numéro 5. Notons que la référence de l'interchange est dans le champ numéro 2.

Pour le segment UNZ (segment de fin d'interchange), le record UNZ001 qui lui correspond conserve la référence de l'interchange dans le champ numéro 2. Notons que pour un même interchange, la référence d'interchange présente dans le record ENV001 doit être la même que celle présente dans le record UNZ001 (ATLAS effectue ce contrôle lors d'une traduction et affiche un message d'erreur en cas d'incohérence).

Le lecteur est invité à ne pas confondre l'étiquette ENV001, qui signale le premier record du fichier plat et qui sera traduit en un segment UNB (début de l'interchange), et l'étiquette INVO01 qui se rapporte au premier record du message (traduit en un segment UNH).

A présent, nous pouvons spécifier le profil des partenaires.

8.2.4. Enregistrement du profil des partenaires

Les partenaires sont de deux types: il y a l'utilisateur (nous) et les parties avec lesquelles cet utilisateur désire échanger des messages. Par conséquent, le logiciel propose l'enregistrement de deux types de profils de partenaires: le nôtre et celui de nos associés.

8.2.4.1. Enregistrement du profil de l'utilisateur

La sélection de l'item *Profile Maintenance* provoque l'affichage d'un sous-menu représenté à la figure 8.18. Comme aucun des réseaux proposés par ATLAS-EDI n'est utilisé ici, leur configuration (Network Profile) est superflue.

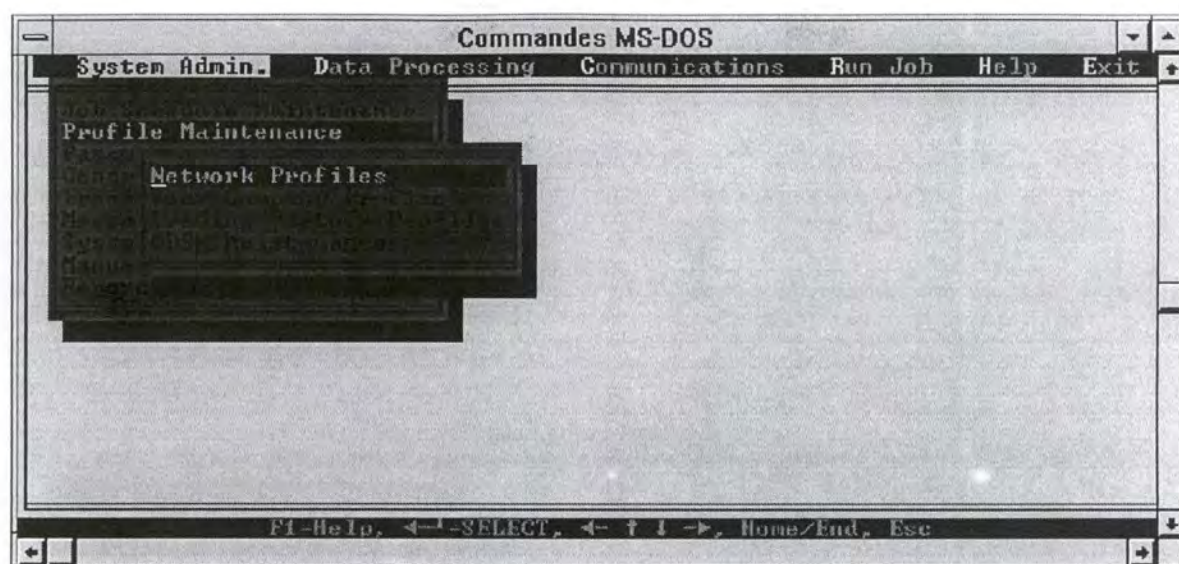


Figure 8.18: Sous-menu de sélection du profil à mettre à jour.

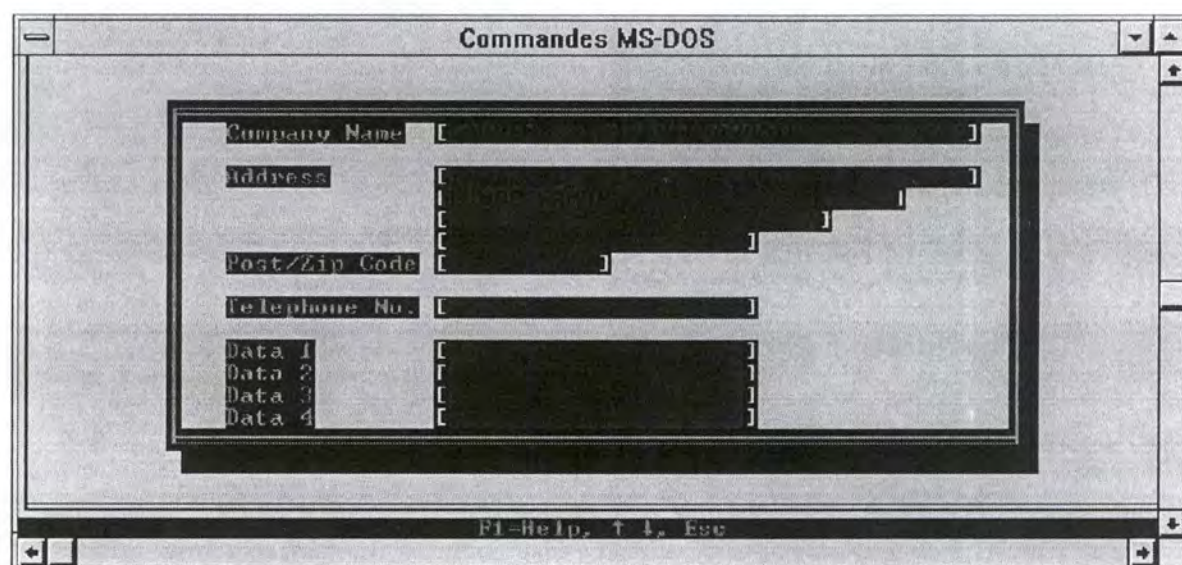


Figure 8.19: Ecran de saisie du profil de l'utilisateur.

Il faut sélectionner l'item *Your Company Profile* et compléter l'écran de saisie qui apparaît. La figure 8.19 illustre l'écran de saisie des informations concernant notre « société ». Dans cet exemple, nous avons écrit les données qui se rapportent à notre simulation. Une fois ces informations enregistrées, elles peuvent être automatiquement sélectionnées pour être incorporées dans les fichiers EDIFACT que l'on désire envoyer. En fait, c'est bien la seule utilité de l'enregistrement du profil de l'utilisateur. Ainsi, les informations que l'on renseigne dans cette section ne doivent pas nécessairement figurer dans les fichiers plats de l'application. Notons bien qu'il s'agit là d'une possibilité offerte à l'utilisateur, et non d'une obligation.

8.2.4.2. Enregistrement du profil des partenaires associés à l'utilisateur

En toute généralité, un utilisateur d'ATLAS-EDI peut entretenir des relations commerciales avec un nombre a priori illimité de partenaires. Ici, nous n'avons qu'un seul partenaire, à savoir Craig Parker de l'Université Monash.

La sélection de l'item *Trading Partner Profiles* illustré à la figure 8.18 provoque l'apparition d'un écran de saisie d'informations. La figure 8.20 montre le contenu de cet écran après que l'on en ait complété les rubriques.

Trader Ref.	0001	
Company Name	CRAIG PARKER	
Address	MONASH UNIVERSITY	
Post/Zip Code	3000	Tel. []
Send Func. Ack.?	NO	Receive Func. Ack.?
Sent Interchng. No.	0000000000	Recd Interchng.
Traders Own Id.	0000000000	
Data 1	[]	Data 2
Data 3	[]	Data 4
Transfer Method	EMAIL	
	Access Code 1	[]
	Access Code 2	[]

Figure 8.20: Ecran de saisie du profil général d'un partenaire commercial.

Ce partenaire est décrit par l'ensemble des informations suivantes:

- sa référence: dans notre système, Craig Parker est référencé par un code qui représente son numéro de partenaire. Il est bon d'insister sur le fait que ce numéro est attribué par nous et qu'il est utilisé à des fins propres à notre organisation.
- nom de sa société (CRAIG PARKER);
- son adresse (MONASH UNIVERSITY, AUSTRALIA);
- les valeurs assignées aux champs Send Func.Ack.? et Receive Func.Ack.? indiquent que l'on ne tient pas compte de la possibilité d'échanger des accusés de réception. En effet, ces champs se rapportent aux situations où des données de type ANSI X12 sont échangées.
- la référence du dernier interchange envoyé à ce partenaire: chaque interchange (ou « enveloppe ») envoyé à un partenaire commercial est identifié par un Interchange Reference Number unique. Ce champ d'édition permet au logiciel d'afficher la référence du dernier Interchange qui a été envoyé à Craig Parker. Ici, comme aucun échange n'a encore eu lieu avec Craig Parker, ce numéro est forcé à zéro.
- la référence du dernier interchange reçu de la part de Craig Parker: ce champ est mis à jour par le logiciel lorsqu'il traduit un interchange en provenance de ce partenaire.

-un identifiant propre à Craig Parker (Traders Own Id.): il s'agit de l'identifiant que le partenaire utilise pour s'auto-identifier. En pratique, l'émetteur d'un interchange EDIFACT doit toujours mentionner son identifiant dans le champ « SENDERS ID » du header prévu à cet effet. De la sorte, le récepteur peut identifier l'émetteur avec certitude. Ainsi, lorsque Craig Parker nous envoie un interchange, il doit mentionner le numéro « 12345 » dans son header. Le SENDERS ID doit donc être connu par lui et par nous préalablement à tout échange EDI. Lors de la traduction IMPORT d'un interchange EDIFACT reçu, le processus de traduction lit la valeur de ce champ afin d'identifier son expéditeur. Si la valeur ne correspond à aucun identifiant d'un des partenaires, alors l'interchange ne peut être traduit.

-quatre champs de données: ces champs peuvent être utilisés pour enregistrer n'importe quel type d'information au sujet du partenaire commercial.

-la méthode utilisée pour échanger les messages avec le partenaire: ce champ doit obligatoirement être spécifié. Comme la méthode de transfert que nous utilisons (le courrier électronique) n'est pas une des méthodes par défaut proposées par le logiciel, nous nous contentons d'indiquer « EMAIL ». Les champs Access Code 1 et Access Code 2 apparaissent à l'écran mais n'ont aucune signification dans ce cas. Normalement, le nom du réseau que l'on mentionne dans ce champ est sélectionné dans une liste de réseaux auxquels le logiciel peut se connecter automatiquement, moyennant l'enregistrement des informations nécessaires à l'établissement de la connexion.

L'écran illustré à la figure 8.20 constitue la première étape de la spécification du profil du partenaire. Il reste à spécifier les informations suivantes:

- le standard et la syntaxe utilisés pour les échanges,
- la version des documents que le partenaire échange.

Ces opérations sont réalisées en sélectionnant d'abord l'item « Documents » de l'écran représenté à la figure 8.20. Un cadre vide apparaît (figure 8.21), comportant les intitulés qui vont servir à identifier le type de message que l'on échange avec ce partenaire:

- le nom du document,
- le nom du message,
- le nom du standard,
- le nom de la syntaxe,
- les données qui identifient la version,
- le numéro de map set qui identifie la table de mapping à utiliser pour réaliser les traductions du message identifié plus haut.

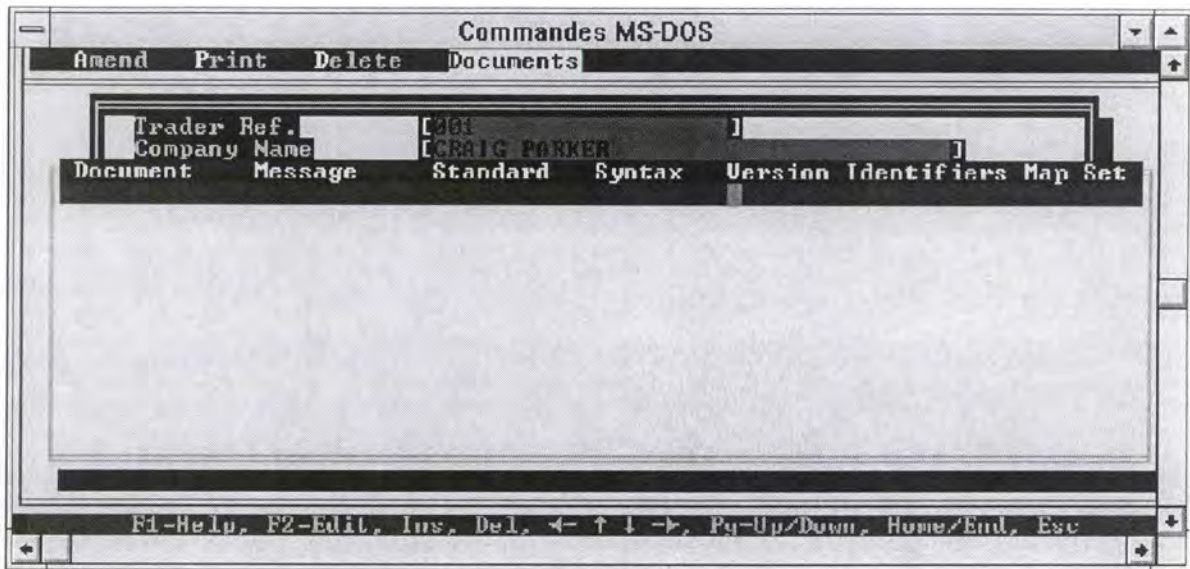


Figure 8.21: Ecran de définition des types de documents échangés par le partenaire commercial.

La figure 8.22 montre quels sont les documents définis que l'on peut sélectionner pour échanger avec le partenaire. Ici, il y en a deux: le document INVOICES et le document ORDERS (qui a été défini auparavant dans un autre but). Il suffit de sélectionner le document INVOICES qui nous intéresse. Ce faisant, le nom du document, accompagné de ses caractéristiques, apparaît en même temps que les documents correspondant aux enveloppes à associer pour constituer un interchange (figure 8.23). Les enveloppes n'ont donc pas besoin d'être sélectionnées, elles sont ajoutées automatiquement dès qu'un document est sélectionné pour le partenaire.



Figure 8.22: Boite de sélection des types de documents échangés par le partenaire commercial.



Figure 8.23: Ecran d'affichage des types de documents échangés par le partenaire commercial.

Tout est prêt à présent pour entamer des traductions du type de message INVOIC. Dans le chapitre 9 nous avons, entre autres, analysé deux traductions concernant ce message INVOIC:

- une traduction IMPORT d'un message EDIFACT correspondant au type défini reçu de notre partenaire (i.e. INVOIC);
- une traduction EXPORT d'un fichier plat en un fichier EDIFACT de type INVOIC à transmettre à notre partenaire.

Comme nous venons de le voir, la configuration du logiciel nécessite de spécifier certains paramètres pour pouvoir effectuer correctement les traductions de fichiers. Parmi les paramètres à spécifier, certains doivent être connus du partenaire et d'autres ne doivent pas l'être. Parmi ceux que le partenaire (Craig Parker) n'est pas censé connaître, on peut citer:

- le format des fichiers plats (importés et à exporter),
- le type des validations effectuées au cours des processus de traduction,
- sa référence dans notre application.

Par contre, il faut que le partenaire connaisse certains paramètres, paramètres déterminés normalement au terme d'une concertation avec lui:

- il doit utiliser les mêmes dénominations de documents que celles renseignées dans son profil;
- il doit nous envoyer des messages conformes à la même version d'EDIFACT que celle que nous utilisons.

De plus, comme les messages que notre partenaire nous envoie doivent être définis dans le profil qu'on a de lui pour être traduits, il doit nous fournir une définition rigoureuse des types de messages qu'il nous envoie et s'y conformer.

Ces remarques s'appliquent également dans l'autre sens.

Chapitre 9. Analyse de la traduction de quelques types de messages

9.1. Traduction d'un interchange EDIFACT en un fichier plat

Dès que la configuration du logiciel est terminée, les traductions peuvent être entreprises. Rappelons que la configuration a été réalisée pour le type de message défini au chapitre précédent.

Il est important de différencier un type de message d'une occurrence de ce type de message.

Les différences possibles entre différentes occurrences d'un même type de message sont:

- des différences de valeurs attribuées aux éléments de données;
- l'existence de certains éléments de données (optionnels), simples ou composites, dans certaines occurrences de messages et pas dans d'autres;
- l'absence de certains segments (optionnels) dans certaines occurrences de messages par rapport aux autres.

En vertu de ces remarques, nous considérons que le message suivant est bien une occurrence du type de message INVOIC défini jusqu'ici (cf. paragraphe 8.1):

```
UNH+1+INVOIC:1:901:UN'
BGM+380:INVOIC+INV001+950706+00'
RFF+PO+PO0001+940810'
NAD+SE+CRAIG PARKER+MONASH UNIVERSITY+++++AU'
CTA+AD+:ACCOUNTING DEPARTMENT'
NAD+BY'
CTA+PD+:PURCHASING PERSON'
LIN+++0300:VN++01:12:TV+1000'
UNT+9+2'
```

En réalité, ce message est reçu sous la forme d'un interchange, comportant les segments de début et de fin d'interchange:

```
UNB+UNOA:1+12345+STUDENTS+950706:1236+REF01++++++1'
UNH+1+INVOIC:1:901:UN'
BGM+380:INVOIC+INV001+950706+00'
RFF+PO+PO0001+940810'
NAD+SE+CRAIG PARKER+MONASH UNIVERSITY+++++AU'
CTA+AD+:ACCOUNTING DEPARTMENT'
NAD+BY'
CTA+PD+:PURCHASING PERSON'
LIN+++0300:VN++01:12:TV+1000'
UNT+9+2'
UNZ+1+REF01'
```

Afin de bien comprendre la signification des segments et des éléments de données qui constituent l'interchange, il est indispensable de se reporter à l'UNTDID (United Nations Trade Data Interchange Directory) de la version 90.1 d'EDIFACT. A ce propos, ATLAS-EDI offre à l'utilisateur la possibilité de consulter le répertoire des segments et des éléments de données d'EDIFACT depuis la version 90.1 jusqu'à la version 93.1.

Nous allons analyser les différents segments qui composent l'interchange à l'aide de ce répertoire.

Voici la liste des segments, accompagnés de leur description et de la description des éléments de données qui les composent.

Le segment UNB:UNOA:1+12345+STUDENTS+950706:1236+REF01++++++1' a comme fonction de débiter, identifier et spécifier un interchange et est interprété de la façon suivante:

- l'identifiant de la syntaxe utilisée est « UNOA » et le numéro de la version syntaxique est égal à 1;
- l'identifiant de l'expéditeur est « 12345 » (il s'agit de l'identifiant utilisé par le partenaire pour s'auto-identifier);
- l'identifiant du receveur est « STUDENTS »;
- la date de la préparation du message est « 950706 », son heure est « 1236 »;
- la référence de contrôle de l'interchange est « REF01 »;
- l'indicateur de test est positionné à 1.

Il faut souligner le fait que, mis à part l'indicateur de test, les éléments de données indiqués par l'expéditeur sont tous obligatoires.

Le segment UNH+1+INVOIC:1:901:UN' représente l'en-tête du message, l'identifie et le spécifie. Il est interprété de la façon suivante:

- le numéro de référence du message est « 1 »;
- l'identifiant du type de message est « INVOIC », son numéro de version est « 1 », son numéro d'édition est « 901 » et l'agence qui le contrôle est « UN ».

Le segment BGM+380:INVOIC+INV001+950706' indique le début du message et contient le numéro qui l'identifie, de même que son type et sa date:

- le code qui correspond au nom du document est « 380 » (380 correspond au Commercial Invoice);
- le nom du document est « INVOIC »;
- le numéro du document est « INV001 »;
- la date du document est « 950706 ».

Le segment RFF+PO+PO0001+940810' constitue la référence de l'INVOIC. Il contient les éléments suivants:

- le qualifiant de la référence est exprimé par le code « PO », ce qui signifie que la facture dont il est question ici se rapporte à un bon de commande adressé par l'acheteur;
- le numéro de référence du bon de commande concerné par la facture est « PO0001 »;
- la date du bon de commande est « 940810 ».

Le qualifiant qui se trouve en première position a comme fonction de donner un sens particulier aux données qui suivent dans le segment. Ici, l'information que le segment apporte est la suivante: la facture (message) se rapporte à un bon de commande daté du 10 août 1994 dont la référence est PO0001.

Le segment NAD+SE+CRAIG PARKER+ MONASH UNIVERSITY+++++AU' spécifie le nom et l'adresse de la partie qualifiée par le premier élément de donnée. Cela signifie que les données qui sont mentionnées dans ce segment ne peuvent s'interpréter qu'en connaissant la signification de la partie à laquelle elles se rapportent. Ici, le qualifiant « SE » correspond à « SELLER », c'est à dire la partie qui a vendu des marchandises à un acheteur. Par conséquent, le nom et l'adresse qui se trouvent dans le segment concernent ce vendeur:

- l'identifiant du vendeur dans cette transaction est «CRAIG PARKER»;
- l'adresse fournie par le vendeur est « MONASH UNIVERSITY »;
- le nom de code du pays du vendeur est « AU », qui correspond à l'Australie.

Le segment CTA+AD+:ACCOUNTING DEPARTMENT' est le segment de contact dont la fonction consiste à identifier une personne (ou un département) à laquelle toute communication devrait être adressée.

Le premier élément de donnée est la fonction de contact (Contact Function) représentée sous la forme d'un code. Le code "AD" contenu dans ce segment spécifie le département comptable (Accounting Department), département auquel il faut adresser nos messages relatifs à la facture reçue. Ce département comptable est nommé "ACCOUNTING DEPARTMENT".

Le segment NAD+BY' doit apporter des informations au sujet de la partie qualifiée par le code "BY" qui signifie "BUYER" (partie à laquelle la marchandise est vendue). Dans ce cas, l'expéditeur n'a pas jugé opportun de renseigner son nom et adresse.

Le segment CTA+PD+:PURCHASING PERSON' est encore un segment de contact qui identifie le "PURCHASING DEPARTMENT" codé par "PD". Ce segment signifie que la communication est adressée au département des achats, et plus précisément à la "PURCHASING PERSON".

Le segment LIN+++0300:VN++01:12:TV+1000' contient des indications à propos du produit ou du service qui fait l'objet de la facture (identification du produit, quantité facturée, prix). Le numéro de l'article facturé est « 0300 », le code qui identifie la catégorie d'articles dont fait partie l'article facturé est « VN » (Vendor Item Code). Le qualifiant de quantité est « 01 » (ce qui signifie qu'il s'agit d'une quantité discrète), la quantité est « 12 » et le spécificateur d'unité de mesure est « TV »(pour Thousand Kilogramms) . Le prix à l'unité s'élève à «1000 ».

Le segment UNT+9+2' est la fin du message (Message Trailer) dont la fonction est de marquer la fin du message et de vérifier son intégrité. En effet, il indique le nombre de segments compris dans le message, "9", et le numéro de référence du message, à savoir "2".

Le segment final UNZ+1+REF01' marque la fin de l'interchange. Il indique la valeur du compteur de contrôle de l'interchange, ici égale à 1, et la référence de contrôle de l'interchange, ici égale à REF01. Remarquons que la référence de l'interchange est bien entendu identique à celle mentionnée dans le premier segment.

Appelons cet interchange R-EMAIL.000, conformément aux conventions adoptées à propos de la dénomination des fichiers. Le « R » signifie que le fichier est « Received », et « EMAIL » traduit le fait qu'il a été reçu via le courrier électronique.

Afin de le traduire sous la forme d'un fichier plat, il suffit de sélectionner l'item *Import Translation* du menu *Data Processing* (figure 9.1). Il en résulte l'apparition dans un cadre d'une liste de fichiers potentiellement traduisibles (figure 9.2). La sélection de R-EMAIL.000 provoque le déclenchement immédiat du processus de traduction IMPORT. Si une erreur survient au cours de la traduction, le processus est interrompu et un message d'erreur apparaît à l'écran.



Figure 9.1: Sélection d'item du menu *Data Processing*.

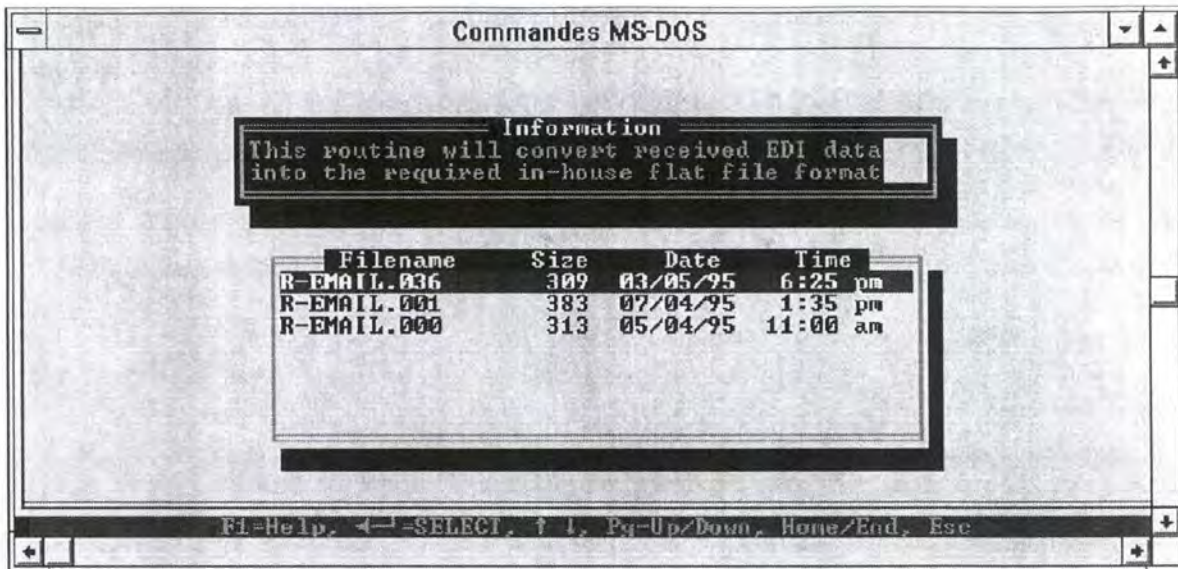


Figure 9.2: Liste des fichiers importables.

Afin de connaître la cause de l'erreur, il faut sélectionner l'item *Audit Trail Report* du menu *Data processing* (figure 9.1) et lire ce que le logiciel y a écrit en espérant trouver l'origine du problème. En cas de succès, le texte contenu dans l'Audit Trail Report est le suivant:

I-Message 296 : Sun Jul 16 23:47:32 1995

***** STARTED IMPORT TRANSLATION *****

Command Line 'd:\atlas2\import.exe R-EMAIL.000 -e0 -C '

I-Message 724 : Sun Jul 16 23:47:33 1995

Cannot find network profile for this data

I-Message 483 : Sun Jul 16 23:47:33 1995

Import Translating 'R-EMAIL.000'

I-Message 299 : Sun Jul 16 23:47:33 1995

Translating an EDIFACT syntax document

Document type is INVOICES

File Position is 0

I-Message 025 : Sun Jul 16 23:47:33 1995

Created New File 'i-invoic.000'

I-Message 323 : Sun Jul 16 23:47:33 1995

Translating interchange with reference REF01

I-Message 297 : Sun Jul 16 23:47:33 1995

***** COMPLETED IMPORT TRANSLATION *****

On peut y lire que le programme import.exe a été appelé depuis le répertoire « D:\atlas2 », à une certaine date et une certaine heure. Le programme import.exe rapporte d'abord le fait qu'aucun profil de réseau n'a été défini, ce qui est normal dans notre cas. Puis, sur base de l'identifiant du partenaire lu dans le segment UNB (« 12345 » à la position 2) et des caractéristiques du message décrites dans le segment UNH, il en déduit que le type de document dont il s'agit est INVOICES. Il s'agit effectivement du nom d'un document répertorié dans le profil de Craig Parker. Tout au long de la traduction, le programme doit valider certains paramètres définis dans les Translation Defaults (Import details). Enfin, dès que l'interchange est complètement traduit, le programme met à jour le profil du partenaire, et plus précisément son champ *Recvd Interchng*. La figure 9.3 illustre ce changement, marqué par l'affichage de la référence de l'interchange (REF01) dans ce champ.



Figure 9.3: Modification de la référence de dernier Interchange expédié dans le profil du partenaire.

Le fichier plat résultant de la traduction se trouve dans le fichier I-INVOIC.000, dont l'indice « I » signifie qu'il est « Imported » et le nom « INVOIC » fait

référence au contenu de l'interchange. Le numéro de version de ce fichier est le même que le numéro du fichier EDIFACT traduit. Voici le fichier plat résultant de la traduction de l'interchange:

```
ENV001^^UNOA^I^12345^^STUDENTS^950706^1236^REF01^^^^^^^I
INVO01^1^INVOIC^1^901^UN^^^^
BGM001^380^INVOIC^INV001^950706^^00^^^^
RFF001^PO^PO0001^^940810^
NAD001^SE^CRAIG PARKER^^MONASH UNIVERSITY^^^^^^^^^^AU
CTA001^AD^^ACCOUNTING DEPARTMENT^^^^^^^^^^
NAD002^BY^^^^^^^^^^
CTA002^PD^^PURCHASING PERSON^^^^^^^^^^
LIN001^^0300^VN^^01^12^TV^1000^^^^^^^^^^
UNT001^9^2
UNZ001^1^REF01
```

Ce fichier est bien constitué d'une séquence de records dont les labels ont été définis au cours du Data Mapping. Les records sont séparés par des sauts de ligne, et les champs par le caractère « ^ », conformément aux indications formulées dans la partie Translation Defaults (Import Details). De plus, le lecteur patient peut comparer le contenu des champs de données du fichier plat et les éléments de données du fichier EDIFACT à l'aide de la table de mapping définie au chapitre précédent pour l'INVOIC.

En regardant ce fichier, il n'est pas difficile de se convaincre de l'intérêt de disposer d'une interface fenêtrée dans Windows, assurant un contexte de lecture convivial. Nous présenterons au chapitre 14 une interface destinée permettant de créer et de modifier de tels fichiers plats.

Dans ce qui suit, nous analysons la traduction d'un fichier plat en un fichier EDIFACT de même type que l'INVOIC.

9.2. Traduction d'un fichier plat en un interchange EDIFACT

L'idée est de transmettre des informations à notre partenaire Craig Parker sous la forme d'un fichier EDIFACT dont l'interprétation est univoque. Dès réception de ce fichier Craig Parker le traduit dans un format qui lui est propre.

Admettons que les informations que nous voulons lui transmettre sont des informations contenues dans notre application. Quelle que soit leur représentation dans l'application, ces données doivent être rassemblées dans un fichier dont le format a été défini lors de la définition des Translation Defaults (Export Details). Ce fichier constitue un fichier plat dont le format est reconnu par le logiciel.

Soit le fichier plat suivant, nommé E-INVOIC.FLT:

```
ENV001^INVOICES^REF02^^001^^1111^^^^^^^^^^^^1
INVO01^1^INVOIC^1^901^UN^^^^
BGM001^380^INVOIC^INV001^950711^^00^^^^
RFF001^PO^PO0012^^950302^
NAD001^SE^STUDENTS^^NAMUR UNIVERSITY^^^^^^^^^BE
CTA001^AD^^ACCOUNTING DEPARTMENT^^^^^^^^
NAD002^BY^^^^^^^^^^^^^^^^
CTA002^PD^^PURCHASING PERSON^^^^^^^^
LIN001^^0200^VN^^01^10^TV^2000^^^^^^^^
UNT001^9^1
UNZ001^1^REF02
```

Les record labels de ce fichier correspondent bien à ceux définis dans la table de mapping de l'INVOIC.

Le premier d'entre eux est fort important puisque le record qu'il identifie contient les éléments suivants:

- en position 1, il contient le nom (INVOICES) du document à traduire (cf. les Export Details);
 - en position 2, on trouve la référence de transmission de l'expéditeur (Senders Transmission Reference), ici égale à « REF02 » (cf. la table de mapping du segment UNB);
 - la référence du partenaire à qui est destinée l'information se trouve en position 5 (cf. les Export Details), il s'agit de la référence de notre partenaire au sein de notre organisation (« 001 ») et non pas de la référence qu'il utilise pour s'auto-identifier (« 12345 »);
 - notre référence de partenaire auprès de Craig Parker est « 1111 », à la position 8 (cf. la table de mapping du segment UNB).
- Notons que l'indicateur de test est présent à la position 20 du fichier plat (cf. la table de mapping du segment UNB).

Analysons brièvement le sens des différents records.

Le record INVO01^1^INVOIC^1^901^UN^^^^ représente l'en-tête du message, il l'identifie et le spécifie. Il s'interprète de la façon suivante:

- le numéro de référence du message est « 1 »,
- l'identifiant du type de message est « INVOIC », son numéro de version est « 1 », son numéro de release est « 901 » et l'agence qui le contrôle est « UN »;

Le record BGM001^380^INVOIC^INV001^950711^^00~~~~~ indique le début du message et contient le numéro qui l'identifie, de même que son type et sa date de création:

- le code qui correspond au nom du document est « 380 »(380 correspond au Commercial Invoice);
- le nom du document est « INVOIC »;
- le numéro du document est « INV001 »;
- la date du document est « 950711 ».

Le record RFF001^PO^PO0012^^950302^ constitue la référence de l'INVOIC. Il contient les éléments suivants:

- le qualifiant de la référence est exprimé par le code « PO », ce qui signifie que la facture dont il est question ici se rapporte à un bon de commande adressé par l'acheteur (Craig Parker);
- le numéro de référence du bon de commande concerné par la facture est « PO0012 »;
- la date du bon de commande est « 950711 ».

Comme dans le cas du message EDIFACT analysé ci-dessus, le qualifiant qui se trouve en première position a comme fonction de donner un sens particulier aux données qui suivent dans le segment. Ici, l'information que le segment apporte est la suivante: la facture (message) se rapporte à un bon de commande daté du 11 juillet 1995 dont la référence est PO0012.

Le record NAD001^SE^STUDENTS^NAMUR UNIVERSITY~~~~~BE spécifie le nom et l'adresse de la partie qualifiée par le premier élément de donnée. Ici, le qualifiant « SE » correspond à « SELLER ». Par conséquent, le nom et l'adresse qui se trouvent dans le record concernent le vendeur, en l'occurrence:

- notre identifiant dans cette transaction est «STUDENTS»,
- l'adresse fournie par le vendeur est « NAMUR UNIVERSITY »,
- le nom de code de notre pays est « BE ».

Le record CTA001^AD^^ACCOUNTING DEPARTMENT~~~~~ contient l'information de contact dont la fonction consiste à identifier une personne (ou un département) à laquelle toute communication devrait être adressée.

Le premier élément de donnée est la fonction de contact (Contact Function) représentée sous la forme d'un code. Le code "AD" contenu dans ce record spécifie le département comptable (Accounting Department), département auquel Craig doit adresser ses messages relatifs à la présente facture. Ce département comptable est nommé "ACCOUNTING DEPARTMENT".

Le segment NAD+BY' doit apporter des informations au sujet de la partie qualifiée par le code "BY" qui signifie "BUYER". Ici, nous n'avons pas voulu indiquer les nom et adresse de notre partenaire.

Le record CTA002^PD^^PURCHASING PERSON^^^^^^^^ est encore un record qui contient l'information de contact. L'information qui s'y trouve signifie que nous nous adressons au "PURCHASING DEPARTMENT" codé par "PD", et plus précisément à la "PURCHASING PERSON".

Le record LIN001^^^0200^VN^^01^10^TV^2000^^^^^^^^ contient des indications à propos du produit ou du service qui fait l'objet de la facture (identification du produit, quantité facturée, prix). Le numéro de l'article facturé est « 0200 », le code qui identifie la catégorie d'articles dont fait partie l'article facturé est « VN » (Vendor Item Code). Le qualifiant de quantité est « 01 » (ce qui signifie qu'il s'agit d'une quantité discrète), la quantité est « 10 » et le spécificateur d'unité de mesure est « TV » (pour Thousand Kilogramms) . Le prix à l'unité s'élève à «2000 ».

Le record UNT001^9^1 est la fin du message (Message Trailer) dont la fonction est de marquer la fin du message et de vérifier son intégrité. En effet, il indique le nombre de segments compris dans le message, "9", et le numéro de référence du message, à savoir "1".

Le record final UNZ001^1^REF02 marque la fin du fichier plat. Il indique la valeur de contrôle « 1 », et la référence de la transmission, ici égale à « REF02 ».

Afin de traduire ce fichier plat sous la forme d'un fichier EDIFACT, il suffit de sélectionner l'item *Export Translation* du menu *Data Processing* (figure 9.1). Comme dans le cas de la traduction Import, il faut choisir dans une liste de fichiers prêts à être traduits celui qui nous intéresse. La sélection de E-INVOIC.FLT provoque le déclenchement immédiat du processus de traduction EXPORT. Si une erreur survient au cours de la traduction, le processus est interrompu et un message d'erreur apparaît à l'écran.

En cas de succès, le texte contenu dans l'Audit Trail Report est le suivant:

E-Message 293 : Mon Jul 17 00:40:50 1995

***** STARTED EXPORT TRANSLATION *****

Command Line 'd:\atlas2\export.exe E-INVOIC.FLT -e0 -C '

E-Message 482 : Mon Jul 17 00:40:50 1995

Export Translating 'E-INVOIC.FLT'

E-Message 025 : Mon Jul 17 00:40:50 1995

Created New File 's-email.034'

E-Message 295 : Mon Jul 17 00:40:50 1995

Creating EDI data for trader 12345

Company name is CRAIG PARKER

Document type being handled is INVOICES

E-Message 323 : Mon Jul 17 00:40:50 1995

Translating interchange with reference REF02

E-Message 294 : Mon Jul 17 00:40:50 1995

***** COMPLETED EXPORT TRANSLATION *****

En lisant ce rapport, on apprend que le programme export.exe a été appelé dans le répertoire D:\atlas2 pour traduire le fichier E-INVOIC.FLT. Au terme de la traduction, le fichier résultant a été nommé « s-email.034 ». Le préfixe « s » signifie que ce fichier est prêt à être « sent », le nom « email » correspond au moyen de transfert spécifié dans le profil du partenaire et le numéro « 034 » est le numéro de version du fichier.

Au cours de la traduction, export.exe lit la référence du partenaire (« 12345 ») destinataire de l'information et en déduit le nom de sa société (CRAIG PARKER). Il reconnaît également le nom du document à traduire et le mentionne (INVOICES). Remarquons que la référence du partenaire et le nom du document à traduire sont les deux informations importantes à spécifier lors de la définition des Export Details (Translation Defaults). Enfin, le logiciel indique dans le rapport la référence contenue dans le premier record et dans le dernier (REF02).

L'interchange EDIFACT issu de la traduction du fichier E-INVOIC.FLT est le suivant:

```
UNB+UNOA:1+1111:21+12345:21+950717:0040+REF02+++++1'
UNH+1+INVOIC:1:901:UN'
BGM+380:INVOIC+INV001+950711+00'
RFF+PO+PO0012+950302'NAD+SE+STUDENTS+NAMUR
UNIVERSITY+++++BE'
CTA+AD+:ACCOUNTING DEPARTMENT'
NAD+BY'CTA+PD+:PURCHASING PERSON'
LIN+++0200:VN++01:10:TV+2000'
UNT+9+1'
UNZ+0+REF02'
```

9.3. Traduction d'un interchange comportant plusieurs messages

Pour rappel, un interchange correspond à une unité d'échange entre partenaires. Jusqu'à présent, les interchanges que nous avons traités ne comportaient jamais qu'un seul message. Dans la pratique, cette situation reste exceptionnelle. En effet, un interchange comporte généralement plusieurs messages de même type. Voici l'interchange que nous nous proposons de traduire:

```
UNB+UNOA:1+12345+EDIGAME:RT+941005:1951+REF3+++++1'
UNH+3+INVOI2:1:901:UN'
BGM+380:INVOIC+INV0001+941005+00'
RFF+PO+PO0001+941003'
NAD+SE+SALES+++SUNNYSIDE TOASTERS:BUILDING E, MONASH
UNIVERSITY+CAULFIELD EAST+VIC+3145+AU'
CTA+AD+:TA'
NAD+BY'
CTA+PD+:BARBIE TARTERS'
FTX+GEN+++COULD YOU PLEASE PAY AS SOON AS POSSIBLE'
LIN+++A000-B001:VN++01:15:EA+2000'
TMA+562500'
UNT+20+3'
UNH+4+INVOI2:1:901:UN'
BGM+380:INVOIC+INV0002+941005+00'
RFF+PO+PO0001+941003'
NAD+SE+SALES+++SUNNYSIDE TOASTERS:BUILDING E, MONASH
UNIVERSITY'
NAD+BY'
CTA+PD+:SOLES MESSIAH'
FTX+GEN+++COULD YOU PLEASE PAY AS SOON AS POSSIBLE'
TMA+607500'
UNT+17+4'
UNZ+2+REF3'
```


Au sein du répertoire d'ATLAS, cet interchange est stocké sous la forme d'un fichier portant le nom R-INTER.003

Cet interchange comporte deux messages de type INVOI2. Ce type de message étant différent de celui que nous avons considéré jusqu'à présent (et que nous appelions INVOIC), nous devons tout d'abord le définir:

Le type de message INVOI2 comprend les segments suivants:

UNH, BGM, RFF, NAD, CTA, NAD, CTA, FTX, LIN, TMA et UNT. Seuls les segments UNH et UNT sont définis comme étant obligatoires, les autres étant déclarés facultatifs. Le segment TMA n'ayant pas encore été décrit dans le type de message précédent, nous en donnons la signification:

Le segment TMA+562500' indique le montant total lié au contenu du message.

Pour pouvoir réaliser la traduction de cet interchange, nous devons tout d'abord effectuer les actions suivantes:

- définir un nouveau type de message à l'aide du Message Table Generator.

- définir une nouvelle table de mapping pour le type de message INVOI2.

- ajouter ce type de message à la liste des messages traités par notre partenaire commercial.

Ces différentes procédures ayant déjà été décrites pour le type de message INVOIC, nous ne les représenterons plus.

Après avoir réalisé la traduction de l'interchange, le texte contenu dans l'Audit Trail Report est le suivant:

I-Message 296 : Sat Jul 15 11:50:57 1995

***** STARTED IMPORT TRANSLATION *****

Command Line 'd:\ATLAS2\import.exe R-INTER.003 -e0 -C '

I-Message 724 : Sat Jul 15 11:50:57 1995

Cannot find network profile for this data

I-Message 483 : Sat Jul 15 11:50:57 1995

Import Translating 'R-INTER.003'

I-Message 299 : Sat Jul 15 11:50:57 1995

Translating an EDIFACT syntax document

Document type is INVOICE2

File Position is 0

I-Message 323 : Sat Jul 15 11:50:57 1995

Translating interchange with reference REF3

I-Message 297 : Sat Jul 15 11:50:58 1995

***** COMPLETED IMPORT TRANSLATION *****

On peut constater que les commentaires fournis par le logiciel restent identiques à ceux qui apparaissent lors de la traduction d'un interchange ne comportant qu'un seul message. Il n'apparaît aucune indication signalant par exemple le nombre de messages traduits. Le type de document reconnu est INVOICE2. Il s'agit en effet du nom attribué au document associé à ce type de message lors de sa définition.

Le nom du fichier créé est I-INVOIC.003

En voici le contenu:

```
ENV001^^UNOA^1^12345^^EDIGAME^^941005^1951^2^^^^^^^1
UNH001^3^INVOI2^1^901^UN^^^^
BGM001^380^INVOIC^INV0001^941005^^00^^^^
RFF001^PO^PO0001^^941003^
NAD001^SE^SALES^^^^^^SUNNYSIDE    TOASTERS^BUILDING    E,
MONASH UNIVERSITY^^CAULFIELD EAST^VIC^3145^AU
CTA001^AD^^TA^^^^^^
NAD002^BY^^^^^^
CTA002^PD^^BARBIE TARTERS^^^^^^
FTX001^GEN^^^COULD YOU PLEASE PAY AS SOON AS POSSIBLE^^^
LIN001^^A000-B001^VN^^01^15^EA^2000^^^^^^
TMA001^562500^^^^
UNT001^20^3
UNH001^4^INVOI2^1^901^UN^^^^
BGM001^380^INVOIC^INV0002^941005^^00^^^^
RFF001^PO^PO0001^^941003^
NAD001^SE^SALES^^^^^^SUNNYSIDE    TOASTERS^BUILDING    E,
MONASH UNIVERSITY^^^^
NAD002^BY^^^^^^
```


CTA002^PD^^SOLES MESSIAH~~~~~

FTX001^GEN^^^COULD YOU PLEASE PAY AS SOON AS POSSIBLE^^^

TMA001^607500~~~~~

UNT001^17^4

UNZ001^2^2

Chapitre 10. Présentation du Mapper et comparaison de son processus de traduction avec celui d'ATLAS

10.1. Introduction

Le Mapper d'Interbridge est un logiciel permettant de convertir un fichier plat en tout autre fichier plat. Un fichier EDI n'étant jamais qu'un fichier plat particulier, le Mapper permet bien évidemment la traduction d'un fichier plat en un fichier EDI.

Ce programme constitue en fait un des éléments de la suite de logiciels Interbridge 5. C'est cependant dans le cadre d'une utilisation isolée que nous avons évalué son utilité dans le monde de l'EDI.

En suivant la classification donnée au chapitre 5, le Mapper peut être considéré comme un convertisseur EDI.

10.2. Interfaces du Mapper

Le Mapper est principalement constitué de deux programmes: le programme IMPBLD qui crée des versions compilées des tables de mapping (les tables établissant la correspondance des données entre fichiers plats), et le programme IMPRUN qui réalise la conversion proprement dite.

Le Mapper ne dispose d'aucun écran de saisie. La communication avec le Mapper s'effectue uniquement à l'aide d'une série de fichiers plats.

La figure 10.1 nous présente les principaux fichiers d'interface du Mapper:

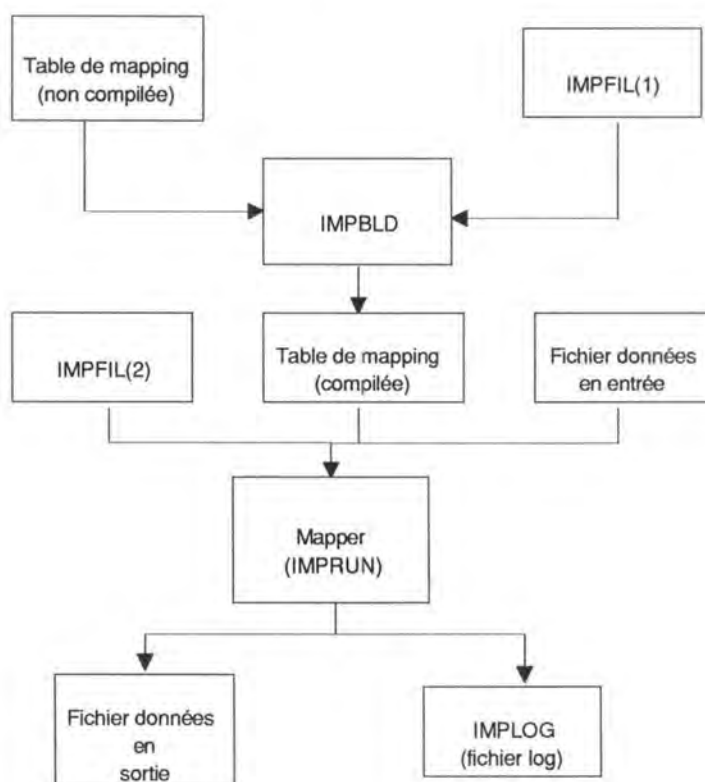


Figure 10.1: Les principaux fichiers d'interface du Mapper

Le principal fichier interface du Mapper est son fichier d'instruction (IMPFIL). Il s'agit d'un fichier texte contenant l'information nécessaire (un scripte) pour réaliser une exécution du Mapper. Dans le cas d'une exécution du programme IMPBLD, il précise notamment le nom du fichier à compiler et le nom du fichier dans lequel sa forme compilée sera stockée. Dans le cas d'une exécution du fichier IMPRUN, il précise, entre autres, le nom de la table de mapping à utiliser et le nom des fichiers de données en entrée et en sortie.

Ce sont bien entendu les tables de mapping qui sont au coeur du processus de traduction offert par le Mapper. Dans la suite du texte, nous allons décrire le format de ces tables, ce qui nous permettra de déduire les fonctionnalités et facilités offertes par ce logiciel. A partir de ces connaissances, nous pourrons établir un comparatif des facilités offertes par le Mapper et ATLAS-EDI.

10.3. Les tables de mapping du Mapper

Sous sa forme non compilée, une table de mapping est un fichier texte constitué d'un ensemble de lignes de longueur variable. Chaque ligne apporte une information. La plupart de celles-ci sont des lignes de données décrivant chacune les positions source et destination d'un élément de donnée déterminé. Certaines lignes ont une fonction particulière et sont appelées lignes d'opération spéciale.

Dans une ligne de donnée, la moitié gauche de la ligne permet de décrire le fichier de sortie, la moitié droite décrivant le fichier en entrée. Les tables sont ordonnées suivant l'ordre des enregistrements du fichier de sortie. La figure 10.2 nous résume le format d'une ligne de donnée.

<i>Fonction</i>	REC NUM	STORE/ RET	STORE ID	DATA START	DATA END	MAND/ COND
<i>Type</i>	(3 numeric)	(1 alpha)	(1 alphnum)	(4 numeric)	(4 numeric)	(1 alpha)
<i>Position</i>	1-3	4	5	6-9	10-13	14

<i>Fonction</i>	DATA FORMAT	DEC PLACES	LITERAL INSERT.	LITERAL START	(Non utilisé)	RECORD ID
<i>Type</i>	(1 alphnum)	(1 numeric)	(5 alphnum)	(4 numeric)		(6 alphnum)
<i>Position</i>	15	16	17-21	22-25	26-32	33-38

Fin du composant Sortie de la ligne de donnée

<i>Fonction</i>	REC NUM	TRIGGER INDIC.	(Non utilisé)	DATA START	DATA END	(Non utilisé)
<i>Type</i>	(3 numeric)	(1 alphnum)		(4 numeric)	(4 numeric)	
<i>Position</i>	39-41	42	43	44-47	48-51	52

<i>Fonction</i>	DATA FORMAT	DEC PLACES	CONVERT CODE	CODE LIST ID	(Non utilisé)	QUALIF. START
<i>Type</i>	(1 alphnum)	(1 numeric)	(1 alpha)	(4 alphnum)		(4 numeric)
<i>Position</i>	53	54	55	56-59	60-63	64-67

Fonction	RECORD QUALIF.	RECORD ID
Type	(3 alphnum)	(6 alphnum)
Position	68-70	71-76

Fin du composant Entrée de la ligne de donnée

Figure 10.2: Format d’une ligne de donnée d’une table de mapping du Mapper

REC NUM

(3 numeric)
1-3 & 39-41

Il s’agit du numéro de l’enregistrement:

- qui va accueillir la donnée dans le fichier en sortie (1-3)
- dont est extrait la donnée dans le fichier en entrée (39-41)

STORE/
RET
(1 alpha)
4

Il s’agit d’un indicateur permettant de sélectionner le rôle joué par ce que l’on appelle, dans le jargon du Mapper, un magasin (store). Un magasin permet d’enregistrer une donnée qui pourra être conservée et réutilisée tout au long du processus de traduction d’un même fichier. Il peut être utilisé pour stocker la donnée extraite du fichier en entrée ou, au contraire, servir de source pour la donnée devant apparaître dans le fichier en sortie.

STORE ID

(1 alphnum)
5

Il s’agit de l’identifiant du magasin susmentionné. En effet, le Mapper permet de manipuler jusqu’à 35 magasins différents.

DATA
START
(4 numeric)
6-9 & 44-47

Il s’agit:

-soit de la position de départ de l’élément de donnée dans le record du fichier de sortie (6-9). Si le fichier de sortie est un fichier dont les éléments de données sont

séparés par un délimiteur (une virgule par exemple), il s'agit du numéro de l'élément recherché.

-soit de la position de départ de l'élément de donnée dans le record du fichier en entrée (44-47). Si le fichier en entrée est un fichier dont les éléments de données sont séparés par un délimiteur (une virgule par exemple), il s'agit du numéro d'ordre de l'élément recherché.

DATA END

(4 numeric)
10-13 & 48-51

Il s'agit de la position finale de l'élément de donnée dans le record du fichier de sortie (10-13) / en entrée (48-51) (pour autant que l'information soit pertinente, c'est à dire que le fichier ne comporte pas de délimiteurs).

MAND/
COND
(1 alpha)
14

Il s'agit de sélectionner le caractère obligatoire (MANDatory) ou optionnel (CONDitional) de l'élément de donnée.

DATA
FORMAT
(1 alphnum)
15 & 53

Cet indicateur permet de sélectionner le format (numérique, alphanumérique ...) de l'élément de donnée en sortie (15) et en entrée (53).

DEC PLACES
(1 numeric)
16 & 54

Permet de sélectionner le nombre de chiffres après la virgule dans le cas d'un élément de donnée de type numérique en sortie (16) ou en entrée (54).

LITERAL
INSERT.
(5 alphnum)
17-21

Permet de définir une séquence de caractères (d'une longueur maximale de 5 caractères) à insérer dans le fichier de sortie. Si l'on veut insérer une séquence plus longue, on le signale en inscrivant dans ce champ le mot-clé « *LT ». Dans ce cas,

la séquence à insérer sera définie par une ligne d'opération spéciale précédant cette ligne de donnée.

LITERAL
START
(4 numeric)
22-25

Il s'agit de la position de départ pour l'insertion de la séquence de caractères dans le fichier de sortie.

RECORD
ID
(6 alphanum)
33-38 & 71-76

Il s'agit d'une étiquette d'identification du record. Dans le fichier en entrée, elle permet d'identifier le record dont on va extraire l'élément de donnée. En sortie, elle permet d'attribuer une étiquette au record en sortie (sa position étant mentionnée dans une ligne d'opération spéciale)

TRIGGER
INDIC.
(1 alphanum)
42

La présence du caractère « * » dans ce champ indique que le record en entrée est un index (trigger) pour une boucle ou un message. Un record index est un record indiquant le début d'un nouveau message ou d'un nouvel élément d'une boucle. Le Mapper permet en effet de définir des boucles de répétition permettant de traduire de façon similaire des données qui se répètent. La définition de la boucle s'effectue grâce aux lignes d'opération spéciale. L'index de message permet au Mapper de connaître la position à partir de laquelle commence un nouveau message (à un message correspond un traitement complet de la table, un nouveau message initialisera donc une nouvelle lecture de la table).

CONVERT
CODE
(1 alpha)
55

Permet de spécifier l'utilisation d'une table de conversion de code. Le Mapper permet en effet de réaliser la conversion d'un code en un autre à l'aide de tables de conversion.

CODE LIST**ID**

(4 alphnum)

56-59

Identifie la liste de code à utiliser au sein du fichier de liste compilé repris dans le fichier d'instruction IMPFIL.

QUALIF. START

(4 numeric)

64-67

Il s'agit de la position de départ du code qualifiant du record en entrée. Le qualifiant permet, en conjonction avec l'identifiant du record, une identification unique de celui-ci. Dans un message EDIFACT, par exemple, l'identifiant de segments de référence sera RFF que la référence soit liée à l'acheteur ou au vendeur. L'utilisation de qualifiants tels que 'BUY' et 'SEL' permet de les différencier.

RECORD**QUALIF.**

(3 alphnum)

68-70

Il s'agit du code qualifiant du record en entrée.

10.4. Les fonctionnalités du Mapper

De l'analyse des tables de mapping, nous pouvons déduire les possibilités de traduction offertes par le Mapper:

- le Mapper permet d'insérer n'importe quel élément de donnée du fichier en entrée à n'importe quelle position dans le fichier de sortie.
- l'identification d'un record peut se faire soit par son numéro, soit par l'utilisation d'un identifiant du record. Un qualifiant peut être précisé pour permettre de différencier deux records possédant un même identifiant.
- les éléments de données peuvent être stockés temporairement au sein de magasins (store). Dans le cas de données numériques, une nouvelle donnée peut être additionnée au contenu d'un magasin. Par la suite, les données stockées dans ces magasins pourront être réutilisées.
- le Mapper permet de définir un caractère obligatoire ou optionnel pour chaque élément de donnée. L'absence d'un élément de donnée obligatoire produira un message d'erreur.
- le Mapper permet de définir des chaînes de caractères à insérer au sein d'un record.
- le Mapper permet de réaliser des conversions entre listes de codes (on peut faire correspondre, pour chaque code d'une liste de code, un nouveau code).
- le Mapper permet de définir des boucles de répétition.

-bien que cela ne soit pas apparu par l'analyse des tables de mapping, le Mapper permet également de réaliser l'entrée manuelle d'informations par l'utilisateur. Typiquement, cette facilité permet de rectifier les principales sources d'erreur apparaissant lors d'un processus de traduction (donnée obligatoire absente, conversion de code impossible, ...)

10.5. Comparaison Mapper - ATLAS-EDI

Nous nous limiterons bien entendu à comparer ce qui peut l'être. Le logiciel ATLAS-EDI comprend de très nombreuses fonctionnalités. Parmi celles-ci, ce sont bien entendu les fonctions d'IMPORT et d'EXPORT qui peuvent être comparées aux fonctionnalités offertes par le Mapper.

10.5.1: Les faiblesses d'ATLAS-EDI vis-à-vis du Mapper

1) Contrairement au Mapper, ATLAS-EDI reste relativement restrictif quant à l'origine des données qui seront stockées dans un record. Prenons l'exemple d'un processus d'IMPORT permettant de convertir un fichier EDI en un fichier plat. Dans ATLAS-EDI:

- La structure de la table de mapping aura obligatoirement pour base la suite des segments définie dans le message EDI. On ne peut pas se baser sur le fichier que l'on veut obtenir en sortie pour réaliser la table de mapping. Une table de mapping est toujours synchronisée sur le message EDI.

- Lors de la traduction, la lecture du fichier à convertir est séquentielle (record par record).

- Les éléments de données d'un même record dans le fichier en sortie proviennent obligatoirement tous d'un seul et même record du fichier EDI (à l'exception des données issues de formules).

En résumé, ATLAS-EDI ne permet pas de panacher au sein d'un même record du fichier sortie des éléments de données provenant de records différents appartenant au fichier en entrée. On conserve toujours l'équation '*un record en entrée <--> un record en sortie*'. Cette limitation peut être particulièrement ennuyeuse lorsqu'une même donnée doit être répétée dans différents records au sein du fichier en sortie ou lorsque des informations provenant de records différents du fichier en entrée doivent être rassemblées au sein d'un même record dans le fichier en sortie.

2) Le concept de magasin (store) n'existe pas chez ATLAS-EDI. Celui-ci est d'ailleurs peu compatible avec les limitations vues en 1).

3) ATLAS-EDI ne permet pas de conversions entre listes de codes.

4) ATLAS-EDI ne permet pas de réaliser l'entrée manuelle directe d'informations par l'utilisateur.

10.5.2. Les avantages d'ATLAS-EDI vis-à-vis du Mapper

Les avantages d'ATLAS-EDI vis-à-vis du Mapper proviennent en premier lieu de la « connaissance » des différents standards EDI que possède ATLAS. Cette « connaissance » facilite tout d'abord la création d'une table de mapping. En effet, rappelons-nous que, sur base d'un message que l'on vient de définir (par

introduction de ses différents segments), ATLAS crée automatiquement une pré-table de mapping comportant l'ensemble des différents éléments de données définis par le standard. L'utilisateur n'a plus qu'à compléter les positions des éléments de données dans le fichier plat et choisir les étiquettes des records du fichier plat pour obtenir une table de mapping valide. De plus, ATLAS offre des possibilités de validation automatique de la syntaxe lors des processus de traduction.

Une autre fonctionnalité importante apportée par ATLAS est la possibilité de définir une série de formules. Ces formules permettent d'introduire automatiquement dans le fichier plat des données telles que l'heure de la transaction, la date, des détails concernant l'expéditeur (sa référence, le réseau qu'il utilise ...), ou encore le résultat de calculs portant sur des variables prédéfinies (par exemple le nombre de segments du message).

10.6. Conclusion

On constate que le Mapper est avant tout destiné à permettre une conversion entre fichiers plats quelconques. Il n'offre aucune facilité propre au monde des standards EDI. Quant à ATLAS, si il offre de nombreux avantages lorsqu'il s'agit d'effectuer une traduction EDI, il se montre finalement très contraignant quant au format des fichiers plats traités. A défaut d'être concurrents, ces logiciels semblent plutôt complémentaires. En effet, le Mapper devrait faciliter le passage vers le monde de l'EDI au sein d'organisations possédant déjà des programmes applicatifs définissant leur propre format de fichiers plats. Si la traduction directe de ceux-ci paraît malaisée à l'aide d'ATLAS, le Mapper devrait permettre une première traduction de ces fichiers plats applicatifs en fichier compatibles avec les possibilités de mapping offertes par ATLAS.

Chapitre 11. Interfaces fenêtrées et applications

Traditionnellement, une organisation commerciale décide de recourir à l'EDI pour améliorer la qualité et la rapidité de ses échanges de documents avec ses partenaires. Ainsi, les documents papiers échangés que sont par exemple les bons de commandes et les factures sont mis sous la forme de fichiers de format EDI. L'entreprise reçoit et émet des documents commerciaux « électroniques » exprimés selon un standard EDI accepté par les partenaires. Comme nous l'avons vu dans la première partie de ce travail, il existe plusieurs formes d'intégration de l'EDI dans les processus d'une entreprise.

Le système que nous avons décrit jusqu'ici est assez rudimentaire puisqu'il n'est pas intégré à une application, par ailleurs inexistante. Le fait que nous ne disposions pas d'application peut sembler étrange au lecteur puisque c'est le besoin de communiquer entre applications qui justifie le recours à l'EDI. Néanmoins, l'existence d'une application n'est pas fondamentale puisque l'objectif principal de ce travail consiste à analyser les étapes à suivre pour établir une relation EDI dans le contexte de la simulation organisée par Craig Parker et Darjan Petric (ces derniers ne disposent pas non plus d'une application).

Afin d'illustrer la possibilité d'interfacer le logiciel ATLAS EDI à d'autres programmes applicatifs, nous avons décidé de concevoir une interface de manipulation de fichiers plats dans un environnement Windows. Cette interface fenêtrée a été développée à l'aide de l'outil Visual Basic 3.0. Par rapport à l'interfaçage des fichiers plats que propose ATLAS EDI, nous disposons ainsi d'un environnement nettement plus convivial.

L'interface est construite sur base de la composition en segments du type de message que nous avons présenté lors de la configuration du logiciel. Elle est donc spécifique à ce type de message et permet d'afficher et d'éditer n'importe quel fichier plat construit selon ce modèle. Comme nous allons le constater, chaque fenêtre de l'interface correspond à un segment du fichier EDIFACT, et chacun de ses champs d'édition contient un élément de donnée. Plus précisément, l'interface permet de visualiser et d'éditer les éléments de données présents dans les fichiers plats correspondant à un type de message au moyen des champs d'édition. Chaque champ d'édition est précédé par un intitulé (exprimé selon la terminologie officielle d'EDIFACT) qui définit l'élément de donnée qui y figure.

La figure 11.1 offre une vue d'ensemble de l'interface:

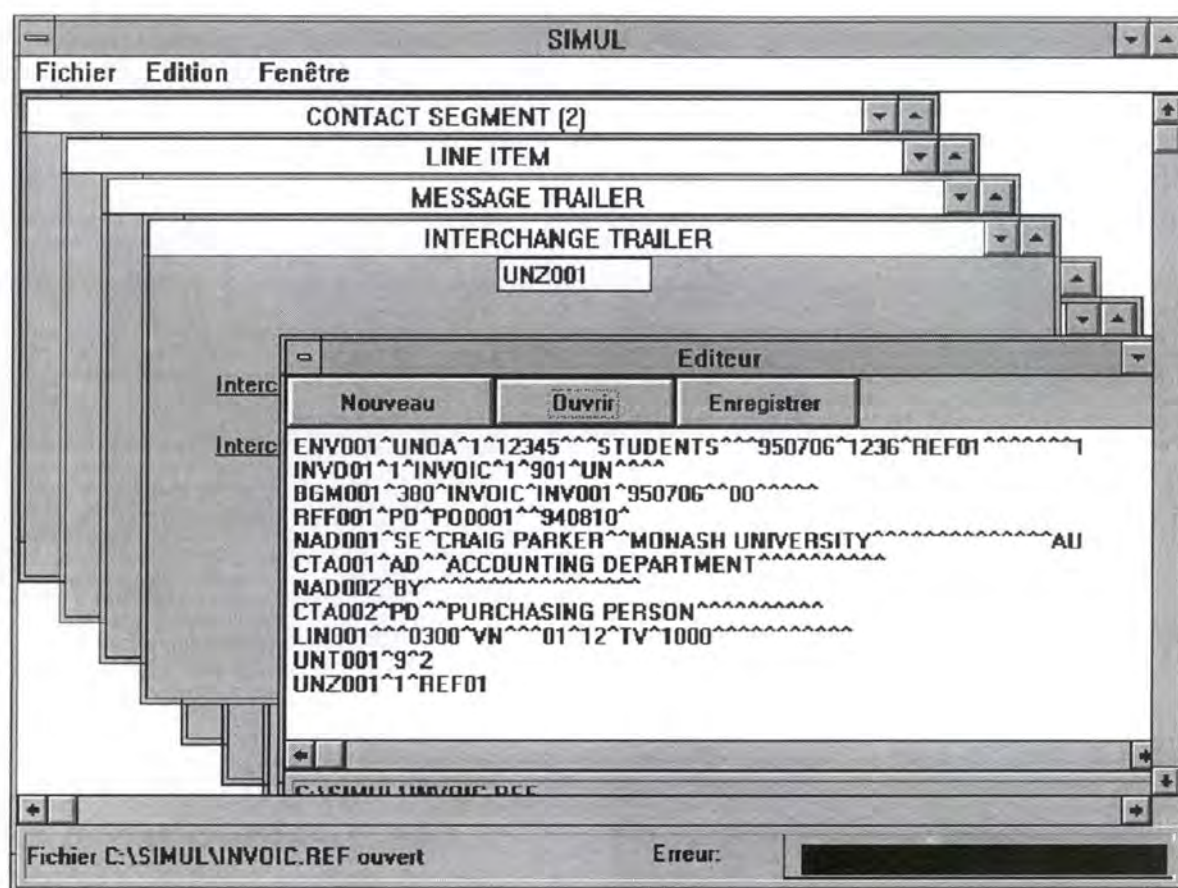


Figure 11.1: Vue d'ensemble de l'interface.

Les items de la barre de menu sont les suivants: « Fichier », « Edition » et « Fenêtre ».

Le menu « Fichier » permet à l'utilisateur:

- de faire apparaître une fenêtre, la fenêtre « éditeur », qui permet de composer manuellement un fichier plat, en indiquant les caractères séparateurs de données et les intitulés des records;
- d'ouvrir un fichier plat existant dans un répertoire quelconque et d'en faire afficher les éléments de données directement dans les fenêtres prévues à cet effet (les caractères séparateurs n'apparaissent évidemment pas);
- d'enregistrer un fichier plat défini au moyen des champs d'édition correspondant aux éléments de données;
- de quitter l'application.

Ainsi, l'utilisateur peut éditer un fichier plat correspondant au type de message étudié en parcourant les différentes fenêtres (représentées en arrière-fond de la figure 11.1) et en éditant les éléments de données qu'il désire.

Le menu « Edition » permet à l'utilisateur de faire afficher directement l'éditeur. L'éditeur permet d'effectuer les opérations suivantes:

- visualiser l'intégralité d'un fichier plat dont les données sont représentées séparément dans les champs d'édition des fenêtres;
- éditer un nouveau fichier plat en utilisant uniquement cette fenêtre d'édition, sans profiter alors des facilités offertes par la disposition ergonomique des champs d'édition;
- éditer n'importe quel fichier de type texte à l'aide de cet éditeur et l'enregistrer dans le répertoire qu'il veut. En particulier, il peut éditer un fichier plat à exporter à l'aide d'ATLAS EDI qui est d'un type différent de celui pour lequel l'interface a été conçue.

Le menu « Fenêtre » permet à l'utilisateur de gérer l'affichage des fenêtres qui correspondent aux records.

Il faut noter que l'édition d'un fichier plat peut se réaliser dans un ordre quelconque des records, puisque les fenêtres peuvent être accessibles et éditées à tout moment. Ce n'est que lorsque l'utilisateur le désire qu'il peut enregistrer les informations qu'ils a renseignées.

La figure 11.2 montre l'intégralité de la fenêtre correspondant au record « MESSAGE HEADER ». Le premier champ d'édition, équivalant ici à INV001, correspond à l'intitulé du second record du fichier plat. Chacun des champs d'édition suivants est précédé de l'intitulé (exprimé selon la terminologie EDIFACT) officiel de l'élément de donnée qu'il peut contenir. Cette propriété a comme conséquence de rendre l'interface utilisable avant tout par des utilisateurs avertis et connaissant bien la norme EDIFACT.

The screenshot shows a window titled "SIMUL - [MESSAGE HEADER]". The menu bar includes "Fichier", "Edition", and "Fenêtre". Below the menu bar, the identifier "INV001" is displayed. The main area contains several fields with labels and values:

Message reference number	1
Message type identifier	INVOIC
Message type version number	1
Message type release number	901
Controlling agency	UN
Association assigned code	
Common access reference	
Sequence message transfer number	
First/last sequence message transfer ind.	

At the bottom, a status bar shows "Fichier C:\SIMUL\INVOIC.REF ouvert" and "Erreur:" followed by a blacked-out area.

Figure 11.2: Fenêtre correspondant à un record particulier du fichier plat

Notons que l'utilisateur peut à tout moment passer d'une fenêtre à l'autre pour y appliquer les manipulations qu'il envisage.

On pourrait imaginer que d'autres programmes, exécutés dans un même environnement Windows, puissent former une application complète qui peut échanger des informations avec d'autres applications éloignées via l'EDI. Par exemple, un utilisateur peut développer une application de gestion qui fonctionne sur micro-ordinateur, et obtenir des fichiers plats « exportables » par ATLAS EDI. De même, cette application pourrait recevoir des fichiers « importés » en provenance d'autres applications installées sur d'autres sites.

Chapitre 12. Méthode de conception des messages échangés

12.1. Choix des messages échangés avec Craig Parker

Jusqu'à présent, nous avons parlé de messages et de types de messages sans spécifier la manière dont nous avons choisi de les construire. Nous savons qu'ils doivent être connus de nous et de Craig Parker pour que chacun d'entre nous puisse configurer son logiciel de façon à pouvoir les traduire correctement.

Idéalement, nous devrions donc définir ensemble les types de messages que nous voulons nous échanger. C'est-à-dire que nous devrions déterminer ensemble:

- la version d'EDIFACT que nous utilisons;
- la séquence des segments constituant les messages;
- le statut de ces segments dans les types de messages (obligatoires ou facultatifs);
- l'existence (et dans l'affirmative procéder à leur définition) de groupes de répétitions des segments.

Dès que ces définitions sont effectuées, on peut parler de types de messages reconnus par les partenaires. Ainsi, les partenaires ne s'échangent que des messages correspondant aux types communément définis. Précisons bien que les messages d'un certain type peuvent différer, par exemple, par la présence (ou l'absence) dans certains d'entre eux de segments facultatifs. Ils peuvent aussi différer par le nombre de répétitions de certains segments (dont le nombre de répétitions est toujours une valeur maximale).

Pratiquement, nous nous sommes mis d'accord avec Craig Parker pour utiliser chaque fois la version 90.1 d'EDIFACT. Les messages que nous avons échangés correspondaient à un type fixé au départ par lui. Nous n'avions qu'à lui demander les caractéristiques du type correspondant aux messages échangés pour pouvoir construire une table de mapping adaptée aux messages. Cette façon de procéder a été la même que dans le cas de la simulation entre les étudiants universitaires d'Australie et de Slovaquie. Les étudiants devaient apprendre les techniques EDI, et les messages qu'ils s'échangeaient étaient soit ceux définis par Craig Parker et Darjan Petric, soit ceux définis par les étudiants eux-mêmes.

D'autre part, comme nous ne disposions pas d'application commerciale, il était peu intéressant de s'échanger une multitude de messages dont l'intérêt ne résidait pas dans leur sémantique. Il a été plus instructif de se concentrer sur quelques types de messages afin d'en étudier les différentes traductions possibles. Nous nous sommes aussi plus intéressés à la méthode à suivre pour négocier la conception des messages. Comme la simulation entre l'Australie et la Slovaquie ne faisait pas intervenir de véritables négociations entre les partenaires commerciaux, nous avons proposé de mieux gérer cet aspect des choses. La meilleure façon d'y parvenir est de se mettre d'accord, préalablement à tout échange, sur la définition des types de messages en s'échangeant des informations sous forme de diagrammes de messages (cf le chapitre consacré à la norme EDIFACT). Cette situation est beaucoup plus

réaliste que celle qui consiste à imposer la structure d'un type de message aux partenaires.

Afin de guider les utilisateurs d'EDIFACT, les Nations unies [UN, 1988] ont publié une méthode de conception des messages qu'il est intéressant de décrire ici à titre de comparaison.

12.2. La conception des messages EDIFACT définie selon les Nations unies

12.2.1. Introduction

Pour rappel, lorsque deux organisations désirent s'échanger de l'information commerciale ou administrative entre leurs systèmes d'information, elles doivent tout d'abord s'accorder sur le contenu et la structure de l'information à échanger.

Au sein du standard EDIFACT, une organisation dispose déjà d'un large éventail de messages électroniques standards et universels pouvant lui permettre d'atteindre ses objectifs. Cependant, lorsqu'une organisation doit, en termes d'échanges d'informations, faire face à des besoins particuliers, il peut s'avérer nécessaire de développer de nouveaux messages. Pour assurer la conformité de ces nouveaux messages au standard EDIFACT, les Nations unies ont émis une série de consignes à respecter lors de leur conception [UN, 1988]. Il est bien évident que ces consignes ont pour objet premier de permettre une intégration cohérente et aisée de ces nouveaux messages au standard préexistant. Cependant, rien n'empêche une organisation de s'en inspirer pour développer ses propres messages qu'elle réserverait à un usage local. Elle dispose ainsi de toute la sécurité et de toutes les facilités liées à l'usage d'un standard éprouvé et largement diffusé.

Ces consignes présentent une série de techniques cohérentes permettant de définir de nouveaux messages ou de modifier des messages préexistants. Elles concernent à la fois la définition du message au niveau de ses éléments de données, de ses segments de données et de sa structure globale. Elles donnent également des indications sur la manière de représenter un message dans un document descriptif.

La suite de ce texte a pour but de présenter l'essentiel de ces consignes.

12.2.2. Principes généraux

Les principes autour desquels s'articulent les indications des Nations unies sont les suivants:

- lors de l'élaboration d'un nouveau message, l'objectif principal est bien entendu de répondre aux véritables besoins de l'organisation, en évitant toute complexité et redondance inutile. Il ne s'agit donc pas nécessairement de produire une copie électronique exacte d'un document papier existant.

- le message doit se conformer au maximum aux recommandations émises par les rapporteurs EDIFACT en ce qui concerne les éléments de données (data elements) utilisés.
- les règles syntaxiques doivent être celles définies dans le document ISO 9735 (UN/ECE EDIFACT Syntax Rules).
- on fera, autant que possible, usage des segments de données standards définis dans le répertoire de l'UN/ECE EDIFACT. Ceux-ci servent toujours de point de départ pour le développement d'un nouveau message.
- les messages et les segments sont développés dans l'optique d'un usage multidisciplinaire. On évite d'en restreindre les possibilités d'utilisation à un domaine particulier.
- la simplicité doit toujours rester une priorité. Inutile de complexifier un message pour gagner quelques caractères à transmettre.

Ces principes généraux sont complétés par une série de recommandations qui concernent le niveau des éléments de données.

12.2.3. Analyse des éléments de données

Si l'on désire créer un nouveau message, il faut tout d'abord identifier l'ensemble des éléments de données nécessaires pour satisfaire la(les) fonction(s) étudiée(s). On recherche ensuite quels sont leurs correspondants au sein du répertoire des éléments de données de l'UNTDID. S'il n'existe pas de correspondant, la demande de définition d'un nouvel élément de donnée standard sera communiquée au rapporteur EDIFACT local.

Pour rappel, un élément de donnée représente la plus petite unité d'information au sein d'un message. Il en existe de deux types, un élément de donnée pouvant être simple ou composé. Un qualifiant est un élément de donnée simple particulier. Il permet de préciser la signification d'un autre élément de donnée. Une liste de code associée au qualifiant permet de donner une signification particulière à l'élément qualifié. Le choix d'une liste de code se fera suivant les mêmes principes que pour un élément de donnée: on utilisera en priorité les listes de codes définies par l'UN/ECE ou, à défaut, un système de codage préexistant défini par un autre organisme de standardisation (l'ISO ou l'IATA par exemple). S'il est nécessaire de définir une nouvelle liste de code, on choisira un format convenant pour l'ensemble des applications attendues et prévisibles de cette liste.

Comme nous allons le voir, les recommandations des Nations unies ciblent naturellement les segments de données.

12.2.4. Analyse des segments de données

Toujours suivant le même principe de cohérence, on privilégiera l'usage des segments de données prédéfinis dans l'EDSD, le répertoire des segments de données standards EDIFACT. Il existe de nombreux avantages à une réutilisation de segments standards, notamment une plus grande rapidité de conception d'un

nouveau message mais également un plus grand conformisme de celui-ci facilitant son usage.

Un nouveau segment devra remplir pleinement ses fonctions et être suffisamment général pour pouvoir être utilisé dans d'autres messages. N'oublions pas qu'un même segment de données reste identique quel que soit le message auquel il appartient. Il contiendra donc toujours la même étiquette et les mêmes éléments de données disposés de la même manière.

En général, on obtient une plus grande flexibilité et un traitement plus aisé en conservant des segments de taille réduite.

12.2.5. Procédure générale d'adoption d'un message

En guise de synthèse, nous rappelons que les messages d'usage universel définis par les Nations unies portent le nom d'UNSM (Universal Standard Messages). Ils sont mis au point et gérés par les rapporteurs EDIFACT et leurs équipes de conseil et de support. La liste de ces messages est reprise dans le EDMD (UN/EDIFACT Message Directory).

Si un groupe d'utilisateurs EDI éprouve le besoin de définir un message EDI à usage universel, ils doivent tout d'abord vérifier qu'il n'existe pas déjà un UNSM permettant de remplir la même fonction.

Si celui-ci existe, mais qu'il ne correspond pas totalement aux attentes des utilisateurs, ceux-ci peuvent proposer une demande de modification du message concerné.

Enfin, si un tel message n'existe pas, ils peuvent présenter une proposition de nouveau message couvrant leurs besoins. Ces propositions de modification ou de création d'un message EDIFACT doivent être adressées au rapporteur UN local. Celui-ci se chargera, avec l'aide des équipes de conseil et de support, d'analyser les propositions avancées. En cas d'approbation par les Nations unies, le nouveau message (ou sa nouvelle version) se verra attribuer un type, un numéro de version et, si nécessaire, un numéro de diffusion (release number).

Chapitre 13. Analyse du chemin suivi par les données

13.1. Introduction

Dans ce chapitre, nous allons détailler le chemin suivi par les données impliquées dans un de nos échanges EDI. L'échange des messages EDI avec notre partenaire australien s'étant réalisé via le réseau Internet, nous commencerons par une introduction portant sur ce réseau. Nous détaillons également la structure du réseau des Facultés et, plus précisément, des différents éléments qui nous permettent une connexion à Internet.

13.2. Le réseau Internet

13.2.1. Introduction

Le terme internet désigne toute collection de réseaux connectés en un réseau plus large de type WAN (Wide Area Network). Ce terme est indépendant des protocoles de communication utilisés par ces réseaux. On peut par exemple avoir des réseaux internet basés sur des protocoles de type AppleTalk ou Netware (Novell). Parallèlement à cette première définition, on désigne sous le terme Internet ou Global Internet un réseau bien particulier connectant des ordinateurs répartis dans le monde entier. Ce réseau, également connu sous le nom de Cyberspace, est un conglomerat regroupant des milliers de réseaux d'ordinateurs. Ceux-ci ont la particularité d'être interconnectés et de partager un même ensemble de protocoles de communication, à savoir les protocoles de la série TCP/IP [RFC 1594].

13.2.2. La structure du réseau Internet

Le réseau Internet actuel est constitué d'un ensemble de sous-réseaux répartis dans le monde entier. Ces réseaux sont reliés par des machines dites passerelles dans le monde IP (« gateway ») ou encore appelées « routeurs IP » qui assurent le transfert de données entre deux sous-réseaux. Les systèmes terminaux (supportant le protocole IP) sont appelés machines hôtes. Les passerelles et les machines hôtes sont identifiées au sein du réseau Internet par leur adresse IP. Une même machine passerelle, appartenant à plusieurs sous-réseaux, peut donc posséder plusieurs adresses IP, une par sous-réseau. Un sous-réseau est, quant à lui, un ensemble logique autonome constitué de supports de transmission et d'équipements actifs de commutation ou d'amplification. Il peut s'agir d'un LAN, d'un MAN ou encore d'un WAN, ou dans une version simplifiée d'une ligne de communication.

13.2.3. Les protocoles TCP/IP

La suite des protocoles TCP/IP [VAN BASTELAER, 1995], du nom des deux protocoles majeurs qui la composent, permet à des ordinateurs de partager des ressources à travers un réseau. On distingue tout d'abord cinq protocoles principaux qui constituent le noyau central de la suite de protocoles TCP/IP.

Il s'agit des protocoles:

- IP (protocole chargé du routage au travers du réseau),
- TCP (fournit une couche Transport en mode connecté),
- FTP (protocole assurant le transfert de fichiers),
- SMTP (protocole soutenant le service de messagerie électronique),
- TELNET (protocole assurant une liaison entre un terminal et une application installée sur une machine hôte distante).

A côté de ces cinq protocoles majeurs, on retrouve d'autres protocoles assurant des facilités additionnelles:

- UDP (fournit une couche transport en mode non connecté)
- SNMP (permet l'échange de messages de gestion du réseau)
- EGP & GGP (protocoles permettant l'échange de messages entre machines passerelles)
- ICMP (permet l'échange de messages entre une machine hôte et une machine passerelle)
- ARP & RARP (permettent la conversion des adresses IP en adresses physiques et réciproquement)
- DNS (supporte la conversion des noms de machine en adresses IP)

La figure 13.1 nous donne la répartition des protocoles majeurs au sein d'une découpe en couches (ou niveaux).

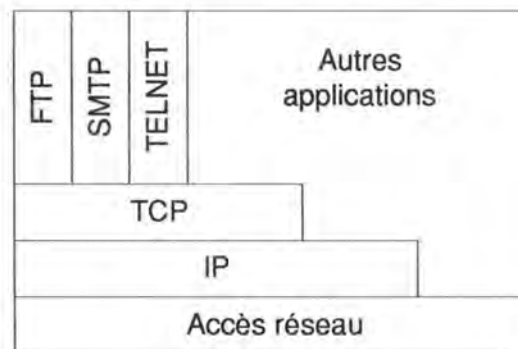


Figure 13.1: Répartition des protocoles majeurs de la série TCP/IP au sein d'une découpe en couches (ou niveaux)

On distingue dans le monde DoD quatre niveaux majeurs :

- la couche d'accès au réseau. Il peut s'agir d'une couche supportant Ethernet, X25 ou un protocole gérant une ligne point à point.
- la couche IP. Elle est chargée de l'acheminement et du routage des paquets de données (appelés datagrammes) au travers du réseau Internet.
- la couche Transport. Celle-ci se charge d'assurer un transport fiable de bout en bout des données au travers du réseau. Elle offre deux protocoles, à savoir TCP et UDP. TCP offre un service en mode connecté. Il assure la réception des données par

le destinataire, sans erreur et dans le respect de l'ordre d'émission. UDP offre quant à lui un service en mode non connecté, sans garantie de bonne réception. Il n'assure pas de contrôle d'erreur ni le respect de l'ordre d'émission des paquets.

-la couche des processus d'Application. Cette couche regroupe l'ensemble des applications faisant appel aux couches inférieures du modèle. On y retrouve notamment les applications offrant les services de transfert de fichiers (FTP), de courrier électronique (SMTP) et de communication entre un terminal et une application sur une machine hôte distincte (TELNET).

Dans la suite de ce document, nous décrivons de manière détaillée le protocole SMTP puisqu'il constitue le support du service de messagerie électronique.

13.2.4. Le protocole SMTP

SMTP [RFC 821; RFC 822] constitue le protocole de base pour le transfert de messages entre ordinateurs sur Internet. SMTP décrit le format du message électronique à échanger (que nous désignerons dans la suite du texte sous le terme de mail) mais également la séquence des messages de contrôle échangés entre deux ordinateurs pour permettre le transfert de ce mail. Voyons cela par un exemple. L'utilisateur qui désire envoyer un mail doit tout d'abord rédiger son mail. Pour cela, il dispose d'une application particulière que nous appellerons l'application de mailing (on parle de local mail User Agent dans la littérature anglophone). Cette application lui permettra également d'examiner les mails qui lui sont envoyés par d'autres utilisateurs du réseau. Généralement, l'interaction entre cette application et le module SMTP se réalise par fichiers interposés. Le fichier créé par l'application de mailing sera récupéré par le module SMTP expéditeur (sender-SMTP module). D'autre part, le module SMTP récepteur (receiver-SMTP module) stockera les messages qu'il reçoit dans ce que l'on appelle couramment une mailbox. Il s'agit le plus souvent d'un fichier ou d'un répertoire privé de l'utilisateur. Suite à une demande provenant de l'application de mailing, le module SMTP expéditeur établit une connexion duplex de bout en bout avec un module SMTP récepteur (à l'aide de TCP par exemple). Celui-ci peut être le récepteur final ou un récepteur intermédiaire. Des commandes SMTP sont alors générées par le module SMTP expéditeur vers ce module SMTP récepteur. Des réponses SMTP sont ensuite expédiées par le module SMTP récepteur vers le module SMTP expéditeur en réponse à ces commandes SMTP.

1) La procédure commence par l'envoi d'une commande MAIL. Cette commande signale au module SMTP récepteur qu'une nouvelle transaction débute. Elle comporte comme argument le <reverse path>. Celui-ci est une liste d'adresses de machines hôtes et de mailbox vers lesquelles des messages d'erreurs peuvent être envoyés. Le premier (et souvent unique) élément de cette liste correspond généralement à l'adresse de la machine hôte expéditrice. Si la transaction est acceptée, une réponse d'accord est expédiée par le module SMTP récepteur.

2) La seconde étape de la procédure est initiée par la commande RCPT. Celle-ci comporte comme argument un <forward-path> identifiant un utilisateur destinataire (parmi ceux sensés posséder une mailbox sur la machine hôte

réceptrice) pour le message. Si le module SMTP récepteur reconnaît ce destinataire, il renvoie une réponse d'accord. Cette seconde étape se répétera autant de fois qu'il existe d'utilisateurs destinataires du message sur la machine hôte destinatrice.

3) La troisième étape correspond à la commande DATA. Si elle est acceptée par le module SMTP récepteur, il renvoie de nouveau un message d'accord. Dès cet instant, toutes les lignes suivantes expédiées par le module SMTP expéditeur sont considérées comme faisant partie du mail. La fin de celui-ci est signalée par une ligne comportant un unique point.

Voici un exemple d'une procédure SMTP fructueuse :

(S : module SMTP expéditeur, R : module SMTP récepteur)

(CRLF : Carriage return and Line Feed)

```
S : MAIL FROM : <lch@info.fundp.ac.be>
R : 250 OK                      message d'accord
S : RCPT TO : <cparker@ponderosa.is.monash.edu.au>
R : 250 OK                      message d'accord
S : DATA R: 354 Start mail input; end with <CRLF>.<CRLF>
S : Blah blah blah...
S : ... etc, etc, etc.
S : <CRLF>.<CRLF>
R : 250 OK                      message d'accord
```

Les commandes MAIL, RCPT et DATA constituent les commandes de base pour l'échange d'un mail. Notons cependant que d'autres commandes, offrant des services additionnels, sont également définies par SMTP.

Le protocole SMTP définit également le format des messages électroniques. Chacun de ceux-ci se compose d'une suite de lignes de texte que l'on peut répartir en deux composants: l' en-tête (« header ») et le corps (« body »).

a) L' en-tête constitue la première partie du message. Créé par l'application de mailing, il est séparé du corps du message par une ligne vide. Chaque ligne de cet en-tête se compose d'un mot-clé (le nom de champ) suivi d'une virgule et de une ou plusieurs données (constituant le corps du champ). Le nom de champ se compose uniquement de caractères ASCII imprimables (i.e. les caractères compris entre 33 et 126 et les décimaux, la virgule exceptée). Le corps du champ peut être composé de n'importe quel caractère ASCII-US (excepté le carriage return et le line-feed (CRLF)).

Voici un exemple de header:

```
Subject: My new email address
To: lch@info.fundp.ac.be, jfh@info.fundp.ac.be
Date: Thu, 23 Feb 1995 16:15:07 +1100 (EST)
From: cparker@ponderosa.is.monash.edu.au (Craig Parker)
Content-Length: 866
```

Certains corps de champs peuvent être interprétés suivant une syntaxe interne. Ces champs sont appelés des champs structurés. Ce sont par exemple les champs reprenant la date et les adresses (To: *adresse(s) de destination*, From: *adresse de l'expéditeur*). D'autres champs, tels que « Subject: », sont reconnus comme de simples chaînes de caractères. De nombreuses lignes peuvent apparaître dans le header d'un message. Certaines sont obligatoires (c'est le cas des quatre premières de l'exemple précédent), d'autres optionnelles. Certaines peuvent même apparaître en plusieurs exemplaires.

b) Le corps du message est constitué de l'ensemble des lignes comprises entre la première ligne vide et la fin du fichier.

L'enveloppe est un élément supplémentaire associé à un message électronique. Elle comporte l'adresse de l'expéditeur et l'adresse du destinataire. Contrairement à ce que l'on peut trouver dans le header du message, l'adresse de destination contenue dans l'enveloppe sera toujours unique. En effet, c'est cette adresse qui sera réellement utilisée pour expédier le message sur le réseau. Si un message est destiné à plusieurs destinataires, on créera autant de copies du message qu'il y a de destinataires. Une enveloppe particulière sera ensuite associée à chaque copie du message avant son expédition.

Dans le cadre de l'étude du protocole SMTP, il nous paraissait intéressant de fournir une introduction à une méthode d'encapsulation de données au sein de messages électroniques. Cette méthode (appelée MIME) offre en effet certaines fonctionnalités propre au monde de l'EDI. Sa description fera l'objet du paragraphe suivant.

13.2.5. L'encapsulation MIME d'objets EDI

13.2.5.1 Introduction

MIME (Multipurpose Internet Mail Extensions) [RFC 1767; RFC 1521; JENKINS, 93] peut être vu comme un mécanisme permettant d'étendre les possibilités de transfert d'information via le système de mail disponible sur Internet. Il fournit à la fois

a) une série de méthodes standards pour l'encodage des informations

b) un mécanisme permettant à une application de spécifier le type d'information contenue dans le mail

a) Le document RFC 822 définissant le format des messages électroniques impose d'importantes contraintes quant au contenu du corps d'un message électronique. En effet, RFC 822 limite ce contenu à un ensemble de lignes de texte codées en caractères ASCII-US.

En conséquence, les utilisateurs dont la langue maternelle nécessite un alphabet plus riche se retrouvent dans l'impossibilité de transmettre des messages syntaxiquement corrects (c'est notamment le cas pour le français puisque les caractères ASCII-US ne comportent pas d'accentuation). De manière plus générale, un mail ne permet normalement pas de transférer des documents non textuels, tels que des images, du son ou des documents multimédias composites. Pour faire face à ces limitations se sont développées une série de techniques permettant de coder les informations non textuelles (ou textuelles mais comportant des caractères non ASCII-US) sous la forme d'une série de caractères ASCII-US. Parmi ces types d'encodage, on retrouve le « quoted-printable », l'« uuencode », le « base 64 », l'hexadécimal pur etc...

b) La deuxième nouveauté apportée par MIME est la possibilité de spécifier au sein du mail le type de données contenues dans le message. Cette spécification est réalisée par l'introduction d'une série de champs supplémentaires au niveau du header du mail. Le document RFC 1521 en définit cinq : Le MIME-Version header-field, le Content-Type header field, le Content-Transfer-Encoding header field, le Content-ID header field et enfin le Content-Description header field.

Le Content-Type header field est l'élément central de MIME. En voici un exemple :

Content-Type: text/plain; charset=iso-8859-1

Un « Content-type header » se divise en trois parties: le « content-type », le « content-subtype » et les paramètres. Dans notre exemple, le « content-type » est « text », ce qui signifie que le contenu du message est textuel. Le « content-subtype » est plain, ce qui signifie que le texte est continu (non entrecoupé de commandes de mise en page par exemple). Quant au paramètre « charset=iso-8859-1 », il identifie la série de caractères utilisée au sein du message.

Les 7 « content-type » standards de base définis dans le document RFC 1521 sont repris dans la figure 13.2.

Content-type	Description des données du mail
text	Information textuelle
multipart	Le mail comprend plusieurs parties, chaque partie pouvant contenir des données d'un type différent. Chacune de ces parties comprend généralement un header permettant de la spécifier.
message	Un message encapsulé. Celui-ci comporte toujours un header répondant aux normes définies dans RFC 822.
image	Des données image
audio	Des données audio
video	Des données vidéo
application	D'autres types de données liées à des applications, typiquement des données binaires.

Figure 13.2: Les 7 « content-type » standards de base définis dans le document RFC 1521

13.2.5.2. MIME et le monde de l'EDI

MIME apporte certaines facilités dans l'échange de documents EDI par mail sur Internet. Comme nous venons de le voir, il permet une sorte d'encapsulation standardisée de divers « objets » au sein d'un message électronique. Ces « objets » peuvent bien entendu être des interchanges EDI. Le document RFC 1767 définit trois « subtype » propres au monde de l'EDI au sein du « content-type » Application. Il s'agit des catégories :

- Application/EDI-X12, indiquant un contenu conforme aux spécifications des standards X12;
- Application/EDIFACT, indiquant un contenu conforme aux spécifications des standards EDIFACT;
- Application/EDI_consent, cette catégorie regroupe tous les autres types de spécifications EDI.

Les avantages apportés par le système MIME sont les suivants:

- bien que les données EDI soient textuelles, certains caractères spéciaux tels que les délimiteurs peuvent parfois être des caractères ASCII non imprimables. Dans le cas

du transfert de données de ce type, le système MIME fournit des systèmes de codage efficaces et standardisés. Ceux-ci seront précisés dans le champ de header Content-Transfer-Encoding.

-en définissant des « subtypes » propres au monde de l'EDI, MIME facilite le travail de l'application de mailing (User Agent) qui serait chargée de l'analyse du contenu des mails reçus par une organisation. Celle-ci pourra en effet reconnaître très facilement le caractère EDI du mail (ou uniquement d'une partie du mail) et transmettre son corps vers les applications chargées de sa traduction au sein de l'organisation réceptrice.

13.2.6. Internet en Belgique: Belnet

Belnet représente l'implantation du réseau Internet en Belgique. Lors de sa création, Belnet avait pour but de permettre l'interconnexion des différents centres de recherche belges et leur connexion avec les réseaux internationaux de même type. A ce jour, il est devenu le support public de base de l'ensemble des services Internet disponibles en Belgique.

Actuellement, le réseau Belnet est basé sur une topologie de réseau en double étoile centrée sur Bruxelles (VUB-ULB) et Leuven (KUL). C'est de ces deux centres que partent respectivement les lignes internationales EUROPANET et EBONE. Le réseau comprend 24 sites connectés plus trois autres sites faisant office de noeuds de concentration. (Louvain-La-Neuve, Anvers et Mons).

13.3. Le réseau des facultés

Les Facultés Notre-Dame de la Paix sont couvertes par un réseau informatique local LAN de type Ethernet supportant les protocoles de la série TCP/IP.

Au sein de l'institut d'Informatique, ce LAN permet de relier deux réseaux Novells aux autres systèmes informatiques. Le premier réseau Novell est composé de 12 machines 486 et d'un serveur nommé Nestor. Le second est composé de 12 P.Cs Pentium tous reliés à un Hub. C'est sur un de ces Pentium (figure 13.3) que nous avons installé notre application ATLAS. Ce microordinateur comporte également un logiciel de mailing Eudora fonctionnant sous environnement Windows 3.11. C'est ce logiciel que nous avons utilisé pour expédier nos mails vers l'Australie.

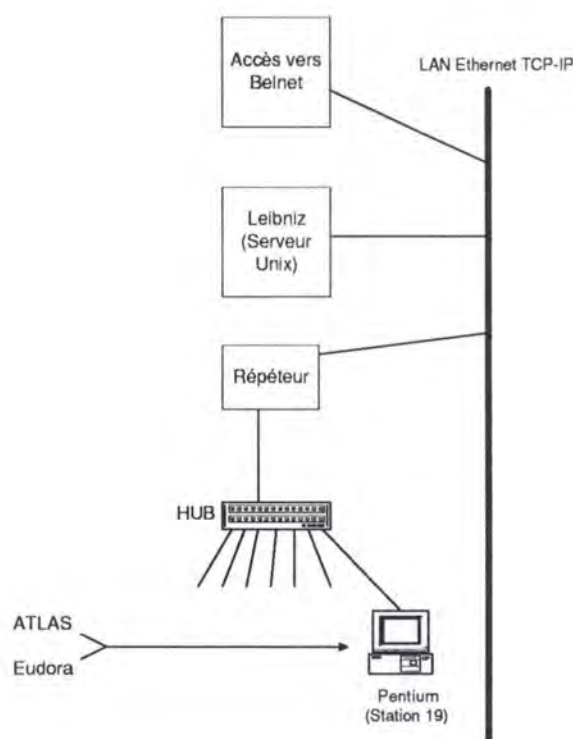


Figure 13.3: Schéma simplifié des éléments du réseau TCP/IP des FUNDP intervenant dans nos échanges

13.3.1. Les échanges entre Eudora et Leibniz

Les échanges entre Eudora et Leibniz [EUDORA, 1994] sont organisés de la manière suivante (cf. figure 13.4) :

-pour les mails sortants:

Eudora utilise le protocole SMTP pour transférer les mails vers le serveur Unix Leibniz. Celui-ci utilise à son tour SMTP pour envoyer les mails sur le réseau Internet.

-pour les mails entrants:

Les mails en provenance d'Internet arrivent tout d'abord sur le serveur Leibniz où ils sont stockés. Lorsque le programme Eudora sera lancé, il consultera la mailbox de Leibniz pour en extraire les mails destinés à son utilisateur. C'est le protocole POP3 (Post Office Protocol) qui est utilisé pour réaliser ce transfert. En effet, SMTP ne fonctionne de manière efficace que si le processus avec lequel il désire établir une connexion reste constamment disponible pour réaliser cette connexion.

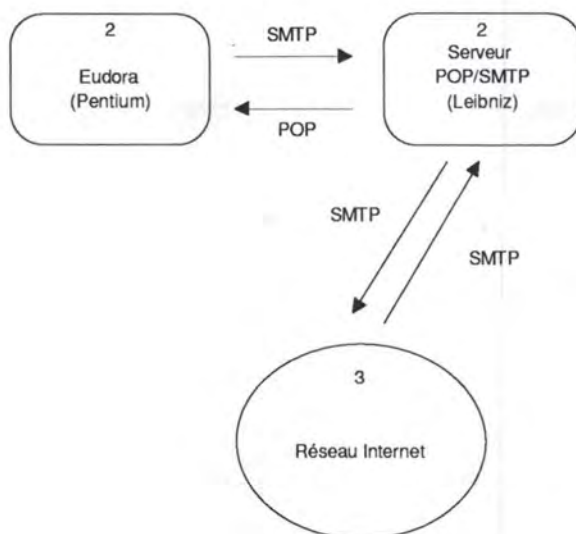


Figure 13.4: Les échanges entre Eudora et Leibniz

Chapitre 14. Synthèse et conclusion

Arrivés au terme de cette analyse de la mise en oeuvre d'une relation EDI au sein d'une organisation (l'institut d'Informatique), nous pouvons en rassembler les idées maîtresses afin d'en tirer les conclusions.

La brève analyse du phénomène EDI a permis d'évoquer les principales caractéristiques de cette technologie sous la forme d'une mise au point destinée à clarifier certains concepts. Ainsi, nous avons notamment rigoureusement défini la notion d'EDI et les types de standards utilisés (standards de représentation et de communication). Comme l'EDI est fondamentalement une technologie au service des organisations, il existe une multitude de possibilités de mise en oeuvre, chacune spécifique à un contexte particulier. C'est la raison pour laquelle il était impérieux d'établir une classification des différents contextes d'intégration de l'EDI.

Sur base de ces éléments, nous avons défini les axes principaux de notre travail, en nous inspirant des résultats de la collaboration entre les Universités de Curtin et de Maribor. Ainsi, tout au long de l'exposé, l'accent a été volontairement placé sur les aspects méthodologiques et techniques de la mise en oeuvre d'une relation EDI. L'originalité de notre travail vient du fait qu'il ne se limite pas à une simple étude théorique sur la mise en oeuvre d'échanges EDI. Nous avons réalisé une première expérience EDI concrète au sein de l'institut d'Informatique de Namur.

Il a été indispensable de présenter une brève description de la norme EDIFACT afin de donner au lecteur les éléments nécessaires à la bonne compréhension de la suite de l'exposé.

Sur base d'échanges de messages EDIFACT et de commentaires avec Craig Parker, nous avons pu évaluer concrètement comment mettre sur pied une relation EDI. Sur le plan technique, notre support logiciel principal était le traducteur ATLAS-EDI.

Nous avons décrit l'ensemble des opérations à effectuer pour configurer ATLAS-EDI. Nous avons également décrit les opérations nécessaires pour réaliser des traductions à l'aide de ce logiciel.

Pour mieux cerner les possibilités d'ATLAS EDI, nous avons aussi réalisé une comparaison de ses fonctionnalités avec celles offertes par un autre logiciel de traduction EDI, le Mapper d'Interbridge.

Ensuite, afin d'illustrer la possibilité d'interfaçage d'ATLAS EDI avec d'autres applications, nous avons également développé une interface de saisie et de présentation de données (développée en Visual Basic) destinée à construire et à lire des fichiers plats correspondant à un type de message EDIFACT.

Comme une large part de notre collaboration avec Craig Parker a été consacrée à concevoir des types de messages EDIFACT, nous avons résumé la façon dont la conception de ces messages a été réalisée. La méthode proposée au départ par Craig

Parker peut être améliorée en attribuant une plus grande importance à la négociation du contenu des messages entre les partenaires.

L'analyse technique des moyens de communication (le courrier électronique) utilisés pour réaliser les échanges EDI nous a éclairés sur le chemin suivi par les messages échangés. Cette analyse pourrait servir de base pour une mise en relation directe du logiciel ATLAS avec le réseau Internet. Elle nous a également permis d'introduire la notion d'encapsulation de message EDI dans un courrier électronique.

Bien entendu, les possibilités de développement de cette étude sont encore vastes.

On peut d'ailleurs dès à présent signaler que Craig Parker envisage de réaliser de nouvelles simulations EDI, caractérisées par un support méthodologique plus rigoureux, notamment en ce qui concerne la négociation de la conception des messages EDIFACT.

Sur le plan technique, on peut aussi proposer une mise en connexion directe du logiciel ATLAS avec le réseau Internet. On peut également imaginer la mise en relation directe d'un traducteur EDI avec une application de gestion (fictive ou non). L'incorporation des facilités offertes par MIME peut également être envisagée. Il va de soi que chacune de ces propositions pourrait faire l'objet d'un nouveau travail.

Bibliographie

[EUDORA, 1994] Eudora Free User Manual, version 1.4.4, Dec 94.

[BERGE, 1991] Berge J., 1991, « The EDIFACT STANDARDS », NCC BlackWell.

[BLOCH, 1991] Bloch S., 1991, « EDI: Echange de Données Informatisé; Introduction à l'échange de données structurées en syntaxe EDIFACT, Tome1 », Editions Eyrolles.

[GEVERS, 1993] Gevers-d'Udekem M., 1993, « Standards EDI de représentation des données », Cahiers de la CITA EDI 7.

[GEVERS, 1994] Gevers-d'Udekem M., 1994, « Classification of EDI STANDARDS and their implications », pp 280-290, The Seventh International Conference - Electronic Data Interchange & Interorganizational Systems, Bled (Slovenia).

[GERSON, 1991] Gerson G.M., 1991, « EDI data base mapping: the key to application integration » EDI Forum, Vol 1, pp 110-117.

[GITS, 1992] Gits N, 1992, « EDI Roadmap », Cahiers de la CITA.

[GITS, 1993] Gits N, 1993, « EDI Roadmap », mémoire de fin d'études, institut d'Informatique, FUNDP, Namur.

[JENKINS, 1993] Jenkins L., 1993, « Exploiting the Internet: EDI as a MIME Content Type », EDI Forum, Vol 6, pp 85-89.

[LEBEAU, 1993] Lebeau B., 1993, « Le rôle de l'EDI dans l'organisation et la restructuration d'une entreprise de messagerie routière: le cas du groupe Ziegler », mémoire de fin d'études, institut d'Informatique, FUNDP, Namur.

[OSITOP, 1991] Ositop, 1991, « EDI & OSI, user options explained ».

[PARKER, 1994] Parker C.M. & Swatman P.M.C., 1994, « The Curtin/Maribor EDI simulation: an innovative approach to teaching international telecommunications », Electronic commerce-Electronic partnership, The Seventh International Conference Electronic Data Interchange & Interorganizational Systems.

[ROSE, 1990] Rose M.T., 1990, « OSI: The Open Book: a practical perspective on OSI », Prentice-Hall.

[RFC 821] Request for Comment: Simple Mail Transfert Protocol.

[RFC 822] Request for Comment: Standard for the format of arpa internet text messages.

[RFC 1521] Request for Comment: MIME (Multipurpose Internet Mail Extensions) Part One : Mechanisms for Specifying and describing the Format of Internet Message Bodies.

[RFC 1594] Request for Comment: Internet: Questions and answers.

[RFC 1767] Request for Comment: MIME Encapsulation of EDI Objects.

[SORTLAND, 1990] Sortland J, 1990, « How to select micro-based EDI translation softwares », EDI Forum, Vol 3, pp 121-124.

[TEDIS, 1990] Bakkenist, « Tedis survey of EDI software products & user requirements », TEDIS, 40 p.

[UN, 1988] Committee on the development of trade - Economic and Social Council - United Nations, 1988, « UN/EDIFACT Message design guidelines », TRADE/WP.4/R.528.

[VAN BASTELAER, 1995] van Bastelaer Ph., 1995, « Téléinformatique et réseaux: matières approfondies », notes de cours, programme de seconde licence et maîtrise en Informatique, FUNDP, Namur.